

DAY3_advent_of_the_cyber

Learning Objectives

- Understanding Authentication
- Understand the use of default credentials and why they're dangerous
- Bypass a login form using BurpSuite

Authentication

Authentication is a process of verifying a users' identity, normally by credentials (such as a username, user id or password); to put simply, authentication involves checking that somebody really is who they claim to be. Authorization (which is fundamentally different to authentication, but often used interchangeably) determines what a user can and can't access; authorization is covered in tomorrow walkthrough, today's task focuses on authentication and some common flaws.

Default Credentials

You've probably purchased (or downloaded a service/program) that provides you with a set of credentials at the start and requires you to change the password after it's set up (usually these credentials that are provided at the start are the same for every device/every copy of the software). The trouble with this is that if it's not changed, an attacker can look up (or even guess) the credentials.

What's even worse is that these devices are often exposed to the internet, potentially allowing anyone to access and control it. In 2018 it was reported that a botnet (a number of internet-connected devices controlled by an attacker to typically perform DDoS attacks) called [Mirai](#) took advantage of Internet of Things (IoT) devices by remotely logging, configuring the device to perform malicious attacks at the control of the attackers; the Mirai botnet infected over 600,000 IoT devices mostly by scanning the internet and using default credentials to gain access.

In fact, companies such as Starbucks and the US Department of Defense have been victim of leaving services running with default credentials, and bug hunters have been rewarded for reporting these very simple issues responsibly (Starbucks paid \$250 for the reported issue):

- <https://hackerone.com/reports/195163> - Starbucks, bug bounty for default credentials.
- <https://hackerone.com/reports/804548> - US Dept Of Defense, admin access via default credentials.

In 2017, it was [reported](#) that 15% of all IoT devices still use default passwords.

[SecLists](#) is a collection of common lists including usernames, passwords, URLs and much more. A password list known as "rockyou.txt" is commonly used in security challenges, and should definitely be a part of your security toolkit.

Dictionary Attacks using BurpSuite

A dictionary attack is a method of breaking into an authenticated system by iterating through a list of credentials. If you have a list of default (or the most common) usernames and passwords, you can loop through each of them in hopes that one of the combinations is successful.

You can use a number of tools to perform a dictionary attack, one notable one being Hydra (a fast network logon cracker) and BurpSuite, an industry-standard tool used for web application penetration testing. Given day 3 is about web exploitation, we'll show you how to use BurpSuite to perform a dictionary attack on a web login form.

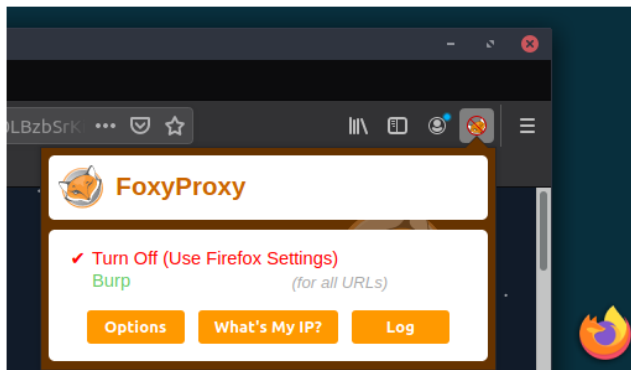
To download BurpSuite click [here](#), otherwise, BurpSuite is pre-installed on our web-based AttackBox.

1. Start BurpSuite, you can do this on the AttackBox by clicking BurpSuite logo in the icon tray.

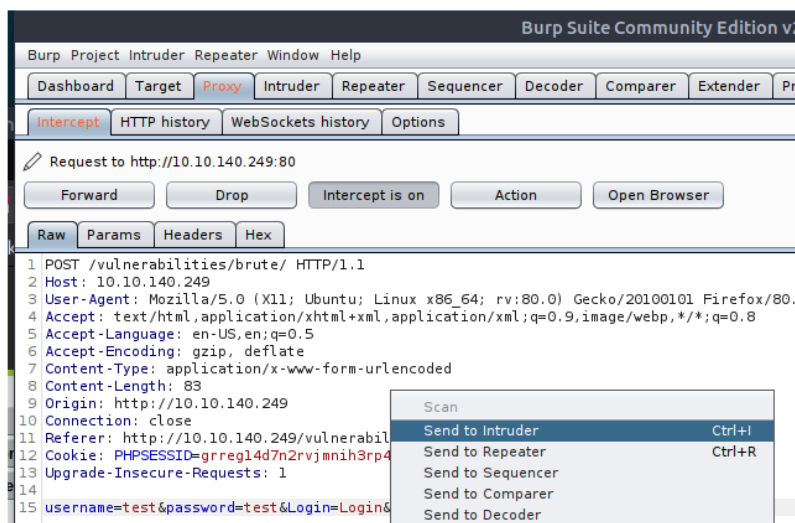


2. Once this has loaded, you want to "Intercept" your traffic by proxying it through the BurpSuite, which will then forward the request to the intended destination (in our case it will be a website) This will give you the ability to analyse and modify your browsers traffic.

1. This example uses the AttackBox, and makes proxying traffic to BurpSuite easy (if you're using BurpSuite on your own machine, click [here](#)) to see how to proxy traffic to BurpSuite). On the AttackBox, open Firefox, click on the FoxyProxy browser extension, and select "Burp" - this will now proxy your traffic to BurpSuite.



2. Go to the BurpSuite application and click the Proxy tab, then click the button "Intercept is on".
3. Navigate to your chosen website, as you're intercepting your traffic, you will see BurpSuite has held your request and will not forward it on until you tell it to. Let's go to our web application and submit your details into a given form, in our case its a generic login form.
4. This captured request will show up in the Proxy tab. Right-click it, and click "Send to Intruder"; BurpSuite has a lot of functionality to repeat modify and manipulate requests, Burp Intruder is a tool to automate customize web attacks. We will use intruder to loop through and submit a login request using a list of default credential, in the hopes that one of the usernames and passwords in the list is correct.



5. Go to the Intruder tab, you should see your request. Here we will insert "positions" (telling Burp which fields to update when automating a request), select a list per position and start the attack.
1. Click the "Positions" tab, and clear the pre-selected positions.
2. Add the username and password values as positions (highlight the text and click "Add")
3. Select "Cluster Bomb" in the Attack type dropdown menu; this attack type iterates through each payloads sets in turn, so every combination of each set is tested.

ⓘ Payload Positions Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```

1 POST /vulnerabilities/brute/ HTTP/1.1
2 Host: 10.10.140.249
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 83
9 Origin: http://10.10.140.249
10 Connection: close
11 Referer: http://10.10.140.249/vulnerabilities/brute/
12 Cookie: PHPSESSID=grrreg14d7n2rvjmnih3rp4ppm2; security=impossible
13 Upgrade-Insecure-Requests: 1
14
15 username=$test$&password=$test$&Login=Login|

```

Add \$ Clear \$ Auto \$ Refresh

6. We're going to tell each "Position" which Payload to use. In our example, we will select a list of usernames for the username field and a list of passwords for the password field.

1. Click the "Payloads" tab, select your Payload set (set 1 is the username field, set 2 is the password field) and add select your list in the "Payload Options" section (or manually add entries).
2. For set 1 (username), we will add a few common default username entries such as "admin", "root" and "user"

Payload set: 1 Payload count: 3
 Payload type: Simple list Request count: 0

ⓘ Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are u

Paste Load ... Remove Clear

Add Add from list ... [Pro version only]

admin
root
user

3. For set 2 (password), we will add a few common default passwords such as "password", "admin" and "12345"

Payload set: 2 Payload count: 3
 Payload type: Simple list Request count: 9

ⓘ Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

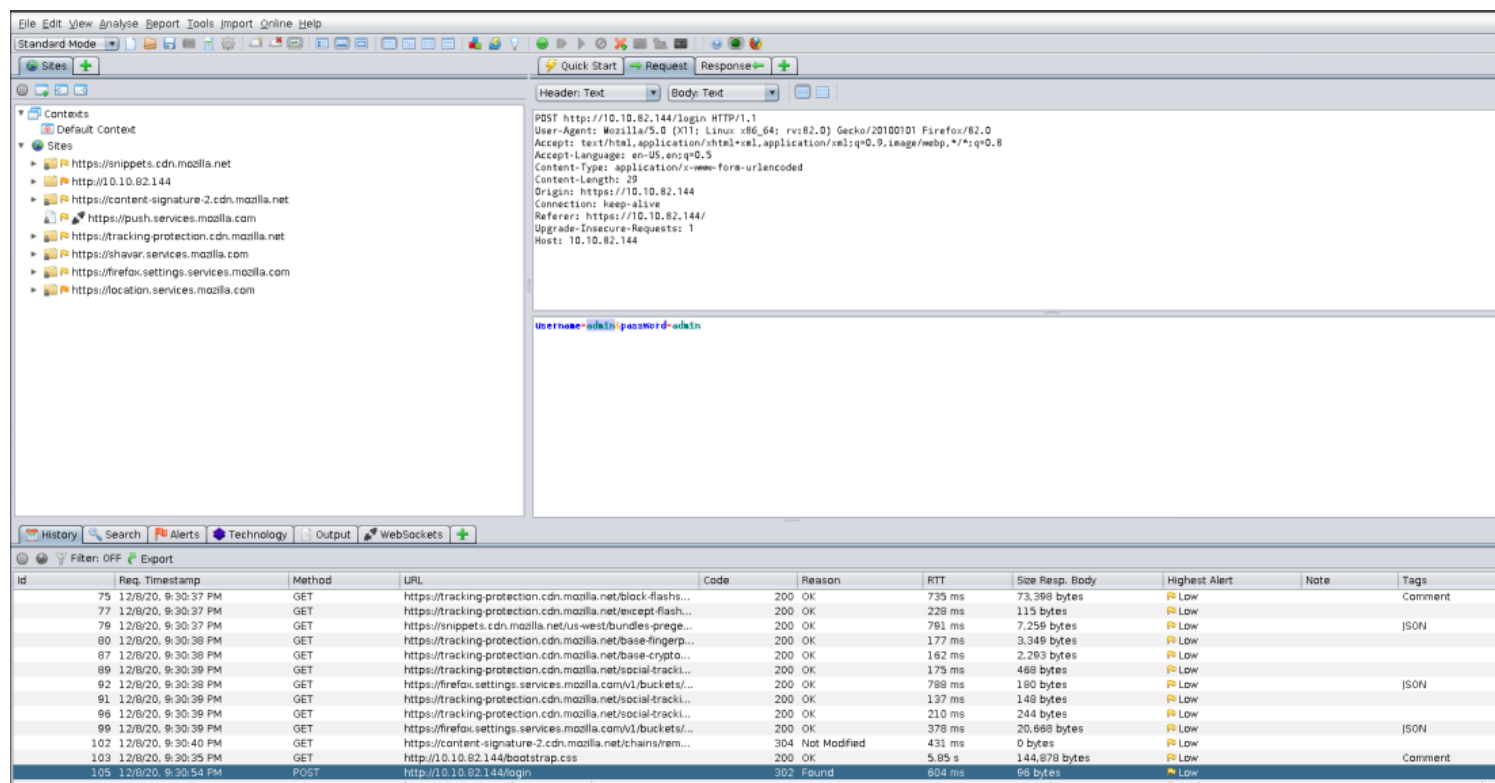
Add Add from list ... [Pro version only]

password
admin
12345

7. Click the "Start Attack" button, this will loop through each position list in every combination. You can sort by the "Length" or "Status" to identify a successful login (*typically all incorrect logins will have the same status or length, if a combination is correct it will be different.*)

Though this challenge has to be done preferably using

BurpSuite, I will use OWASP ZAP(due to various limitations in community edition of BurpSuite).



on seeing the request tab in above screenshot, we can see the attempted “username” and “password”. we will now use fuzzer(after highlighting and adding string/list one by one for each) to try various username and passwords combinations.

To get the correct combination we can see ‘response size’, ‘state’ , etc. Here all response sizes are 96 bytes , except one(60 bytes) .

On seeing a response with size of 96 bytes, we can see the response to reach the conclusion that these combinations are not the correct ones .

Quick Start Request Response

Header Text Body Text

Context

```

ps://snippets.cdn.mozilla.net
p://10.10.82.144
ps://content-signature-2.cdn.mozilla.net
https://push.services.mozilla.com
ps://tracking-protection.cdn.mozilla.net
ps://shavar.services.mozilla.com
ps://firefox.settings.services.mozilla.com
ps://location.services.mozilla.com

```

```

POST http://10.10.82.144/login HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: https://10.10.82.144
Connection: keep-alive
Referer: https://10.10.82.144/
Upgrade-Insecure-Requests: 1
Host: 10.10.82.144

```

```

username=root&password=root

```

Search Alerts Technology Output WebSockets Fuzzer

Progress: 1: HTTP - http://10.10.82.144/login 100% Current fuzzers: 0

| Message Type | Code | Reason | RTT | Size Resp. Header | Size Resp. Body | Highest Alert | State | Payloads |
|--------------|------|--------|--------|-------------------|-----------------|---------------|-----------|--------------|
| 0 Original | 302 | Found | 604 ms | 218 bytes | 96 bytes | Low | | root, root |
| 1 Fuzzed | 302 | Found | 628 ms | 218 bytes | 96 bytes | | | root, 12345 |
| 3 Fuzzed | 302 | Found | 600 ms | 218 bytes | 96 bytes | | | admin, admin |
| 5 Fuzzed | 302 | Found | 603 ms | 218 bytes | 96 bytes | | | admin, 12345 |
| 6 Fuzzed | 302 | Found | 226 ms | 200 bytes | 60 bytes | | | user, root |
| 7 Fuzzed | 302 | Found | 255 ms | 218 bytes | 96 bytes | | Reflected | user, admin |
| 8 Fuzzed | 302 | Found | 264 ms | 218 bytes | 96 bytes | | Reflected | user, 12345 |
| 9 Fuzzed | 302 | Found | 266 ms | 218 bytes | 96 bytes | | Reflected | admin, root |
| 4 Fuzzed | 302 | Found | 1.59 s | 218 bytes | 96 bytes | | | root, admin |
| 2 Fuzzed | 302 | Found | 1.68 s | 218 bytes | 96 bytes | | | |

Quick Start Request Response

Header Text Body Text

Context

```

ps://snippets.cdn.mozilla.net
p://10.10.82.144
ps://content-signature-2.cdn.mozilla.net
https://push.services.mozilla.com
ps://tracking-protection.cdn.mozilla.net
ps://shavar.services.mozilla.com
ps://firefox.settings.services.mozilla.com
ps://location.services.mozilla.com

```

```

HTTP/1.1 302 Found
X-Powered-By: Express
Location: /?login=username_incorrect
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 96
Date: Tue, 08 Dec 2020 16:05:41 GMT
Connection: keep-alive

```

```

<p>Found. Redirecting to <a href="/?login=username_incorrect">/?login=username_incorrect</a></p>

```

Search Alerts Technology Output WebSockets Fuzzer

Progress: 1: HTTP - http://10.10.82.144/login 100% Current fuzzers: 0

| Message Type | Code | Reason | RTT | Size Resp. Header | Size Resp. Body | Highest Alert | State | Payloads |
|--------------|------|--------|--------|-------------------|-----------------|---------------|-----------|--------------|
| 0 Original | 302 | Found | 604 ms | 218 bytes | 96 bytes | Low | | root, root |
| 1 Fuzzed | 302 | Found | 628 ms | 218 bytes | 96 bytes | | | root, 12345 |
| 3 Fuzzed | 302 | Found | 600 ms | 218 bytes | 96 bytes | | | admin, admin |
| 5 Fuzzed | 302 | Found | 603 ms | 218 bytes | 96 bytes | | | admin, 12345 |
| 6 Fuzzed | 302 | Found | 226 ms | 200 bytes | 60 bytes | | | user, root |
| 7 Fuzzed | 302 | Found | 255 ms | 218 bytes | 96 bytes | | Reflected | user, admin |
| 8 Fuzzed | 302 | Found | 264 ms | 218 bytes | 96 bytes | | Reflected | user, 12345 |
| 9 Fuzzed | 302 | Found | 266 ms | 218 bytes | 96 bytes | | Reflected | admin, root |
| 4 Fuzzed | 302 | Found | 1.59 s | 218 bytes | 96 bytes | | | root, admin |
| 2 Fuzzed | 302 | Found | 1.68 s | 218 bytes | 96 bytes | | | |

For the combination with the response size of 60 bytes,

HTTP/1.1 302 Found
X-Powered-By: Express
Location: /tracker
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 60
Date: Tue, 08 Dec 2020 16:05:41 GMT
Connection: keep-alive

Found. Redirecting to tracker

| Message Type | Code | Reason | RTT | Size Resp. Header | Size Resp. Body | Highest Alert | State | Payloads |
|--------------|------|--------|--------|-------------------|-----------------|---------------|-----------|--------------|
| 0 Original | 302 | Found | 604 ms | 218 bytes | 96 bytes | Low | | root, root |
| 1 Fuzzed | 302 | Found | 628 ms | 218 bytes | 96 bytes | | | root, 12345 |
| 3 Fuzzed | 302 | Found | 600 ms | 218 bytes | 96 bytes | | | admin, admin |
| 5 Fuzzed | 302 | Found | 603 ms | 218 bytes | 96 bytes | | | admin, 12345 |
| 6 Fuzzed | 302 | Found | 226 ms | 200 bytes | 60 bytes | | Reflected | user, root |
| 7 Fuzzed | 302 | Found | 255 ms | 218 bytes | 96 bytes | | Reflected | user, admin |
| 8 Fuzzed | 302 | Found | 264 ms | 218 bytes | 96 bytes | | Reflected | user, 12345 |
| 9 Fuzzed | 302 | Found | 266 ms | 218 bytes | 96 bytes | | | admin, root |
| 4 Fuzzed | 302 | Found | 1.59 s | 218 bytes | 96 bytes | | | root, admin |
| 2 Fuzzed | 302 | Found | 1.68 s | 218 bytes | 96 bytes | | | |

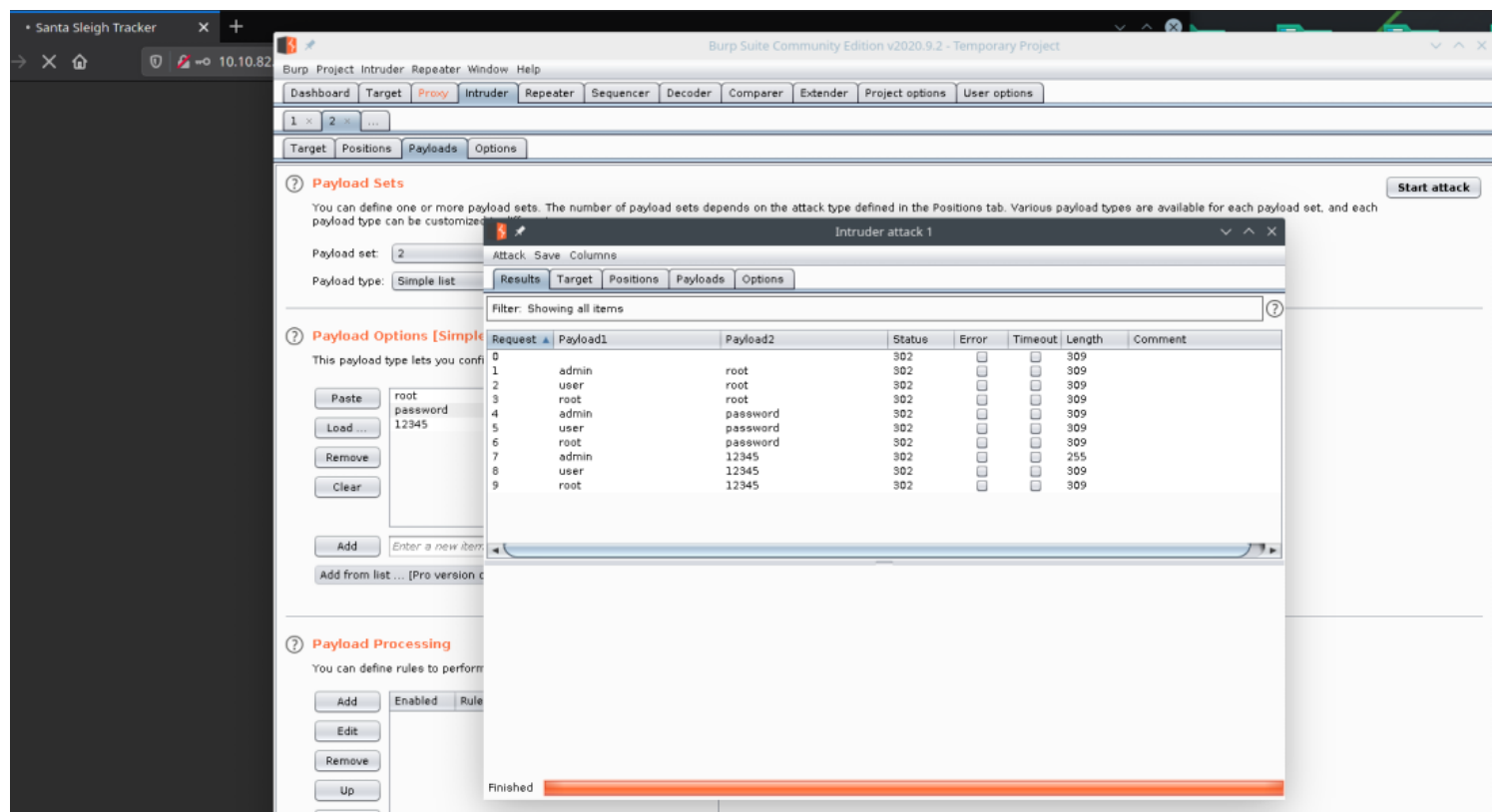
POST http://10.10.82.144/login HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: https://10.10.82.144
Connection: keep-alive
Referer: https://10.10.82.144/
Upgrade-Insecure-Requests: 1
Host: 10.10.82.144

user=admin;password=12345

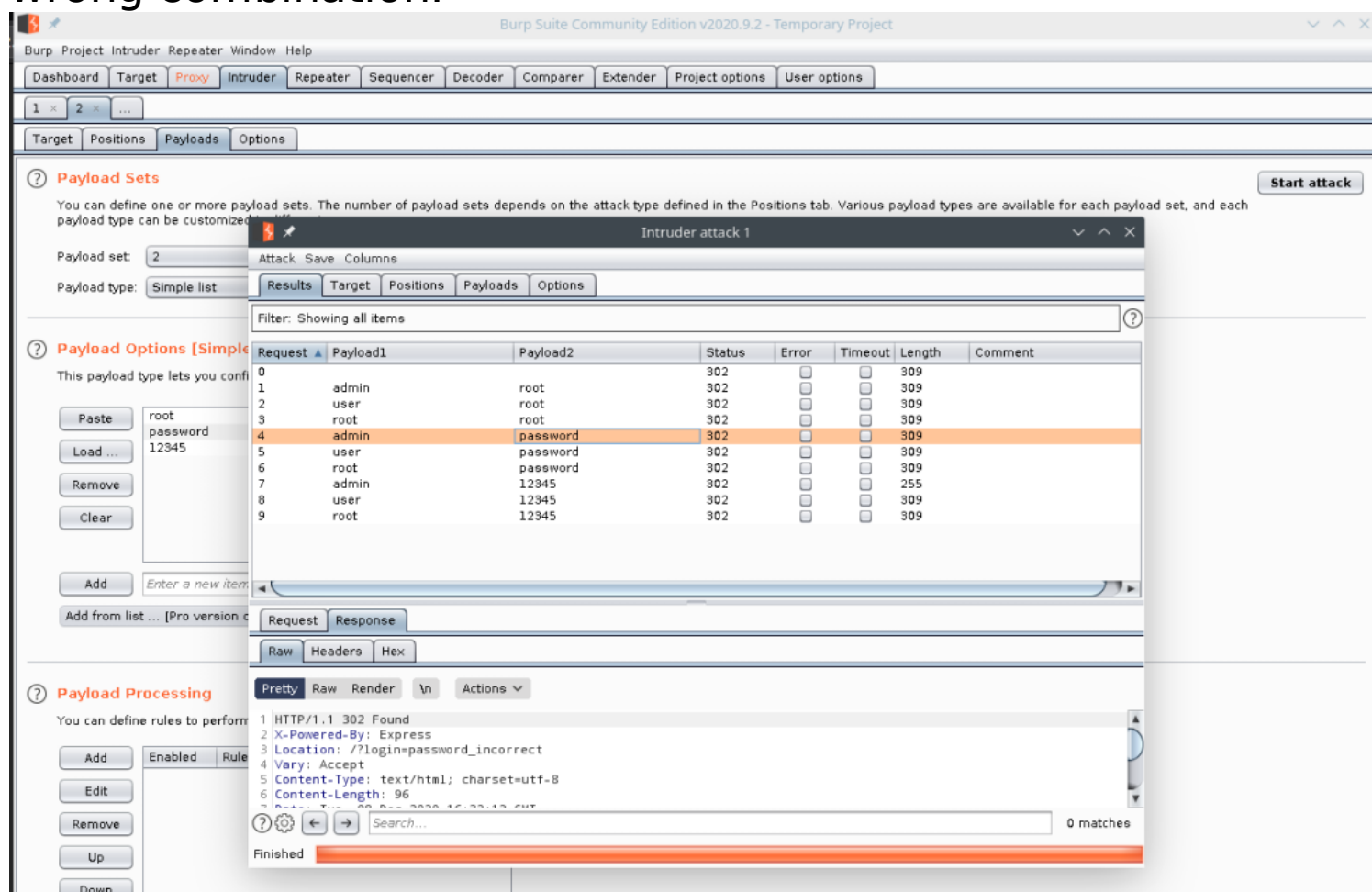
| Message Type | Code | Reason | RTT | Size Resp. Header | Size Resp. Body | Highest Alert | State | Payloads |
|--------------|------|--------|--------|-------------------|-----------------|---------------|-----------|--------------|
| 0 Original | 302 | Found | 604 ms | 218 bytes | 96 bytes | Low | | root, root |
| 1 Fuzzed | 302 | Found | 628 ms | 218 bytes | 96 bytes | | | root, 12345 |
| 3 Fuzzed | 302 | Found | 600 ms | 218 bytes | 96 bytes | | | admin, admin |
| 5 Fuzzed | 302 | Found | 603 ms | 218 bytes | 96 bytes | | | admin, 12345 |
| 6 Fuzzed | 302 | Found | 226 ms | 200 bytes | 60 bytes | | Reflected | user, root |
| 7 Fuzzed | 302 | Found | 255 ms | 218 bytes | 96 bytes | | Reflected | user, admin |
| 8 Fuzzed | 302 | Found | 264 ms | 218 bytes | 96 bytes | | Reflected | user, 12345 |
| 9 Fuzzed | 302 | Found | 266 ms | 218 bytes | 96 bytes | | | admin, root |
| 4 Fuzzed | 302 | Found | 1.59 s | 218 bytes | 96 bytes | | | root, admin |
| 2 Fuzzed | 302 | Found | 1.68 s | 218 bytes | 96 bytes | | | |

so , we have a correct combination as we can see in response .

Doing the same thing we did above(using burpsuite):



wrong combination:



right combination:

Burp Suite Community Edition v2020.9.2 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized.

Start attack

Payload set: 2 Attack Save Columns

Payload type: Simple list Results Target Positions Payloads Options

Filter: Showing all items

Payload Options [Simple list]

This payload type lets you configure a list of payloads.

Paste root password 12345

Load ...

Remove

Clear

Add Enter a new item

Add from list ... [Pro version only]

| Request | Payload1 | Payload2 | Status | Error | Timeout | Length | Comment |
|---------|----------|----------|--------|--------------------------|--------------------------|--------|---------|
| 0 | | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 1 | admin | root | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 2 | user | root | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 3 | root | root | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 4 | admin | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 5 | user | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 6 | root | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 7 | admin | 12345 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 255 | |
| 8 | user | 12345 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |
| 9 | root | 12345 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 309 | |

Request Response

Raw Headers Hex

Pretty Raw Render \n Actions

1 HTTP/1.1 302 Found
2 X-Powered-By: Express
3 Location: /tracker
4 Vary: Accept
5 Content-Type: text/html; charset=utf-8
6 Content-Length: 60
7 Date: Tue, 20 Dec 2020 16:33:15 GMT

Search... 0 matches

Finished