Vulnerability Analysis

Applications

- niltalk
- gitness
- Incubator Answer

Vulnerabilities

The search for weaknesses will target the following vulnerabilities:

Title	OWASP Code	Description
Path Confusion	4.2.13	Read code where the server paths are defined. Check for badly configured paths, usualy with improper usage of regular expressions.
Directory Traversal File Include	4.5.1	Many web applications use and manage files as part of their daily operation. Using input validation methods that have not been well designed or deployed, an aggressor could exploit the system to read or write files that are not intended to be accessible. In particular situations, it could be possible to execute arbitrary code or system commands.
JWT	4.6.10	Check the algorithm used, signature validation, weak secret key used in the HMAC (if applicable), private information leakage, and improper check of the claims defined in RFC 5719.
Stored Cross Site Scripting	4.7.2	Stored Cross-site Scripting (XSS) is the most dangerous type of Cross Site Scripting. Web applications that allow users to store data are potentially exposed to this type of attack.
Malicious File Upload	4.10.9	Identify features with file inclusion, check for malware upload, XML file upload, and filter invasion.

niltalk

niltalk was tested, running locally at localhost:9000.

4.2.13 - Path Confusion

> *"... replace all the existing paths with paths that do not exist, and then examine the behavior and status code of the target."*
>
> OWASP

First technique was to *fuzz* the main path, in search for some hidden path that may had misplaced by the developers. This was done, firstly with `dirb` and it's default wordlists, and then with `wfuzz` with common list from SecLists.

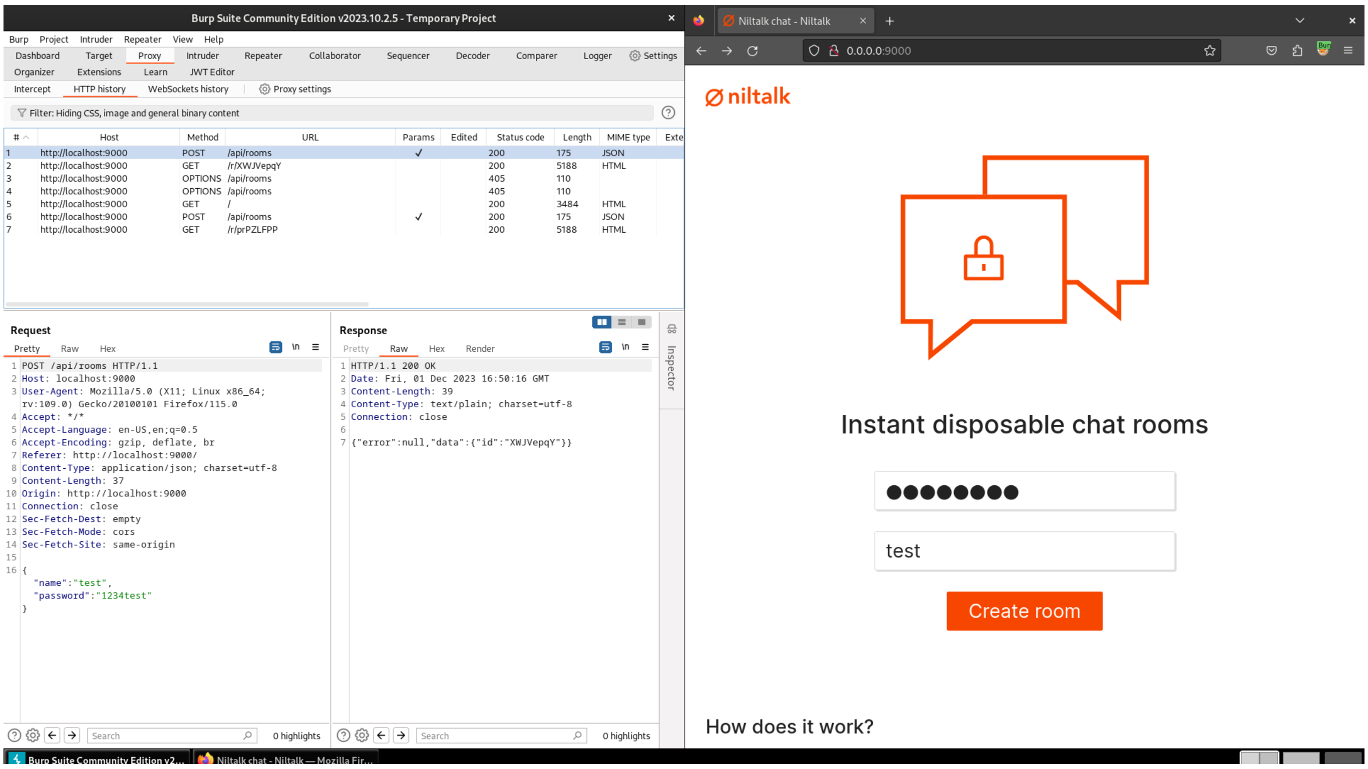
```
root@parrot:/usr/share/dirb/wordlists$ for file in $(ls -fA); do dirb
http://localhost:9000/ $file; done
```

```
root@parrot:/usr/share/wordlists/seclists$ wfuzz -c -z file,Discovery/Web-
Content/directory-list-1.0.txt --sc 200 http://10.0.2.2:9000/FUZZ
```

No matches were found fuzzing the application.

By reading the documentation for this particular application tho, we learn that each chat room is given an auto-generated and unique id, which is used to create the URL path for that room. And being that each room is ephemeral and with a very limited timeout, exposed path are improbable to occur.

On the other hand, what appears to be infrastructure endpoints were discovered when intercepting traffic. Such as the `api/rooms` endpoint.


Burp Suite Community Edition v2023.10.2.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn JWT Editor
Intercept HTTP history WebSockets history Proxy settings
Filter: Hiding CSS, image and general binary content
Niltalk chat - Niltalk
0.0.0.0:9000
niltalk
Instant disposable chat rooms
●●●●●●●●●
test
Create room
How does it work?

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Exte
1	http://localhost:9000	POST	/api/rooms	✓		200	175	JSON	
2	http://localhost:9000	GET	/r/XWJVepqY			200	5188	HTML	
3	http://localhost:9000	OPTIONS	/api/rooms			405	110		
4	http://localhost:9000	OPTIONS	/api/rooms			405	110		
5	http://localhost:9000	GET	/			200	3484	HTML	
6	http://localhost:9000	POST	/api/rooms	✓		200	175	JSON	
7	http://localhost:9000	GET	/r/prPZLFPP			200	5188	HTML	

Request

```
POST /api/rooms HTTP/1.1
Host: localhost:9000
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://localhost:9000/
Content-Type: application/json; charset=utf-8
Content-Length: 37
Origin: http://localhost:9000
Connection: close
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{
  "name":"test",
  "password":"1234test"
}
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 01 Dec 2023 16:50:16 GMT
Content-Length: 39
Content-Type: text/plain; charset=utf-8
Connection: close

{"error":null,"data":{"id":"XWJVepqY"}}
```

An extended fuzzing discovery technique was attempted at this path, but it didn't resolve any undisclosed endpoint. This iterated over multiple dictionaries, fuzzing with each one.

```
path="/usr/share/wordlists/seclists/Discovery/Web-Content/api"; for file in
$(ls -fA $path); do wfuzz -c --sc 200 $path/$file -u
http://localhost:9000/api/FUZZ; done
```

However, testing for wrong paths of possible secret endpoints does not expose a weakness in the application.

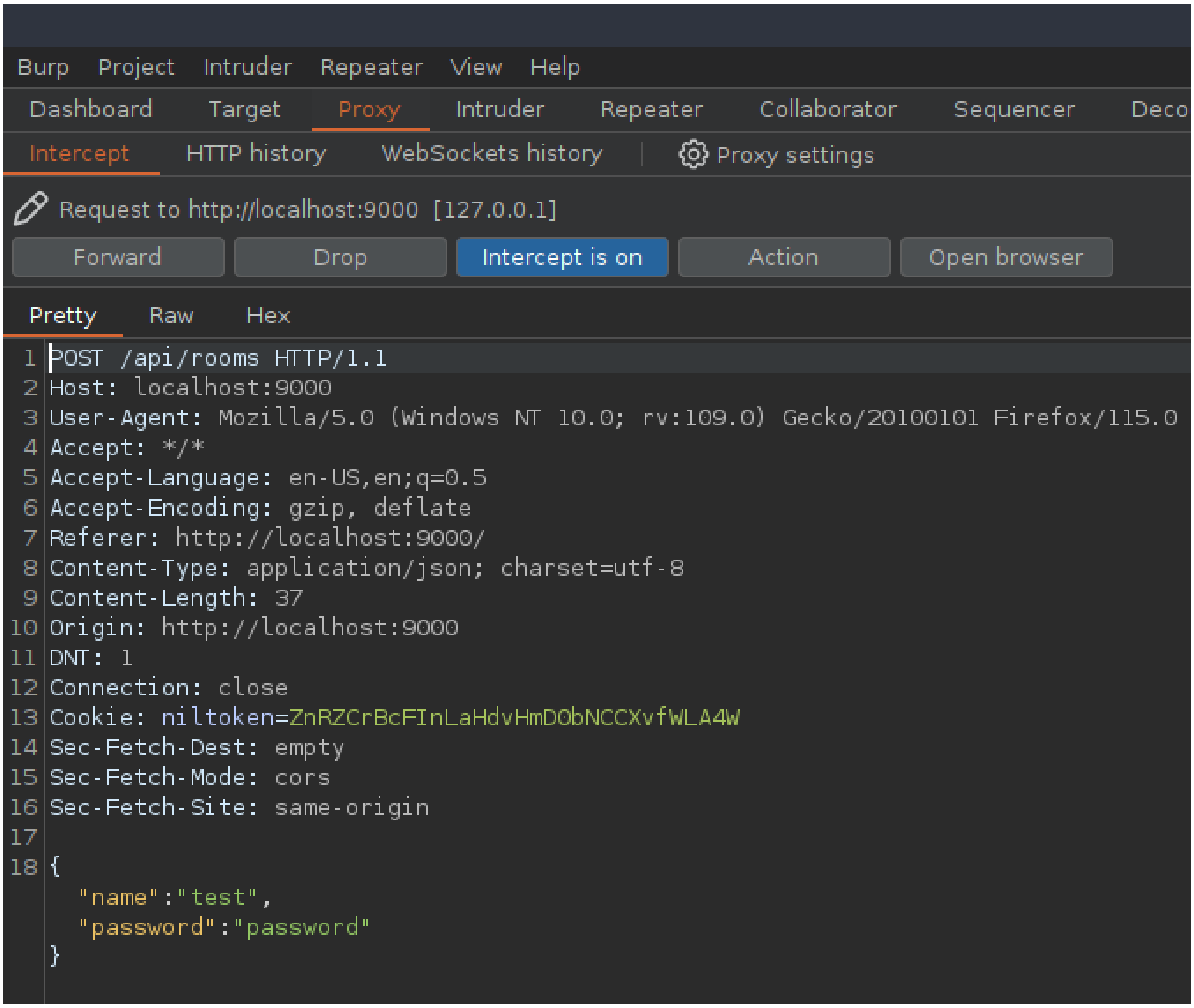
4.5.1 - Directory Traversal File Include

> *"... enumerate all parts of the application that accept content from the user."*
>
> OWASP

niltalk being a simple chat room application, does not offer much in terms of user interaction beside written message. Nonetheless, the three main interaction that require user input are:

1. Room creation
2. Entering and authenticating into a room
3. Write message

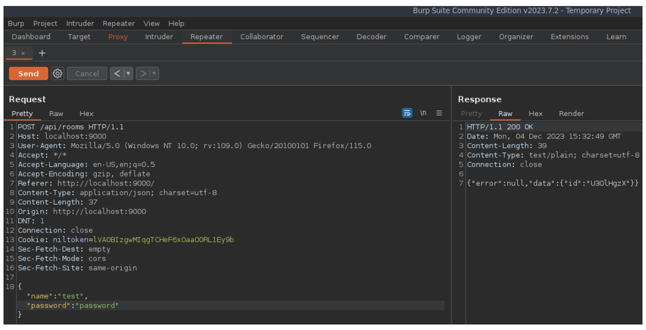
To create a new room, a user must provide a `name` for the room, and a `password` that will be needed for users to access that room.


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Deco
Intercept HTTP history WebSockets history Proxy settings
Request to http://localhost:9000 [127.0.0.1]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

```
POST /api/rooms HTTP/1.1
Host: localhost:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9000/
Content-Type: application/json; charset=utf-8
Content-Length: 37
Origin: http://localhost:9000
DNT: 1
Connection: close
Cookie: niltoken=ZnRZCrBcFInLaHdvHmDObNCCXvfWLA4W
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{
  "name":"test",
  "password":"password"
}
```

Interestingly, collision avoidance for room with sames name is implemented, so it is not clear how the room ids are created.

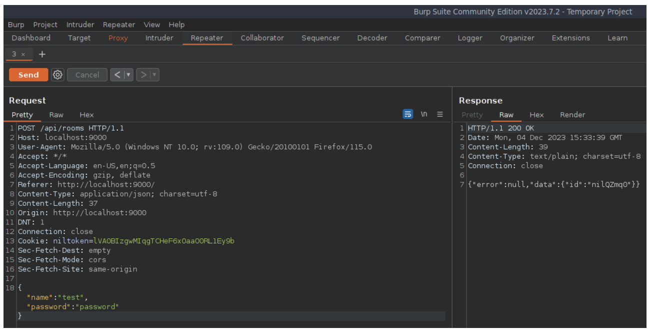


```
POST /api/rooms HTTP/1.1
Host: localhost:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9000/
Content-Type: application/json; charset=utf-8
Content-Length: 37
Origin: http://localhost:9000
DNT: 1
Connection: close
Cookie: niltoken=lVA0BIzgwMIqgTCHeF6x0aa00RL1Ey9b
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{
  "name":"test",
  "password":"password"
}
```

```
HTTP/1.1 200 OK
Date: Mon, 04 Dec 2023 15:32:49 GMT
Content-Length: 39
Content-Type: text/plain; charset=utf-8
Connection: close

{"error":null,"data":{"id":"U3OlHgzX"}}
```



```
POST /api/rooms HTTP/1.1
Host: localhost:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9000/
Content-Type: application/json; charset=utf-8
Content-Length: 37
Origin: http://localhost:9000
DNT: 1
Connection: close
Cookie: niltoken=lVA0BIzgwMIqgTCHeF6x0aa00RL1Ey9b
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{
  "name":"test",
  "password":"password"
}
```

```
HTTP/1.1 200 OK
Date: Mon, 04 Dec 2023 15:33:39 GMT
Content-Length: 39
Content-Type: text/plain; charset=utf-8
Connection: close

{"error":null,"data":{"id":"nilQZmq0"}}
```

4.6.10 - JWT

4.7.2 - Stored Cross Site Scripting

4.10.9 - Malicious File Upload

gitness

4.2.13 - Path Confusion

4.5.1 - Directory Traversal File Include

4.6.10 - JWT

4.7.2 - Stored Cross Site Scripting

4.10.9 - Malicious File Upload

Incubator Answer

4.2.13 - Path Confusion

4.5.1 - Directory Traversal File Include

4.6.10 - JWT

4.7.2 - Stored Cross Site Scripting

4.10.9 - Malicious File Upload