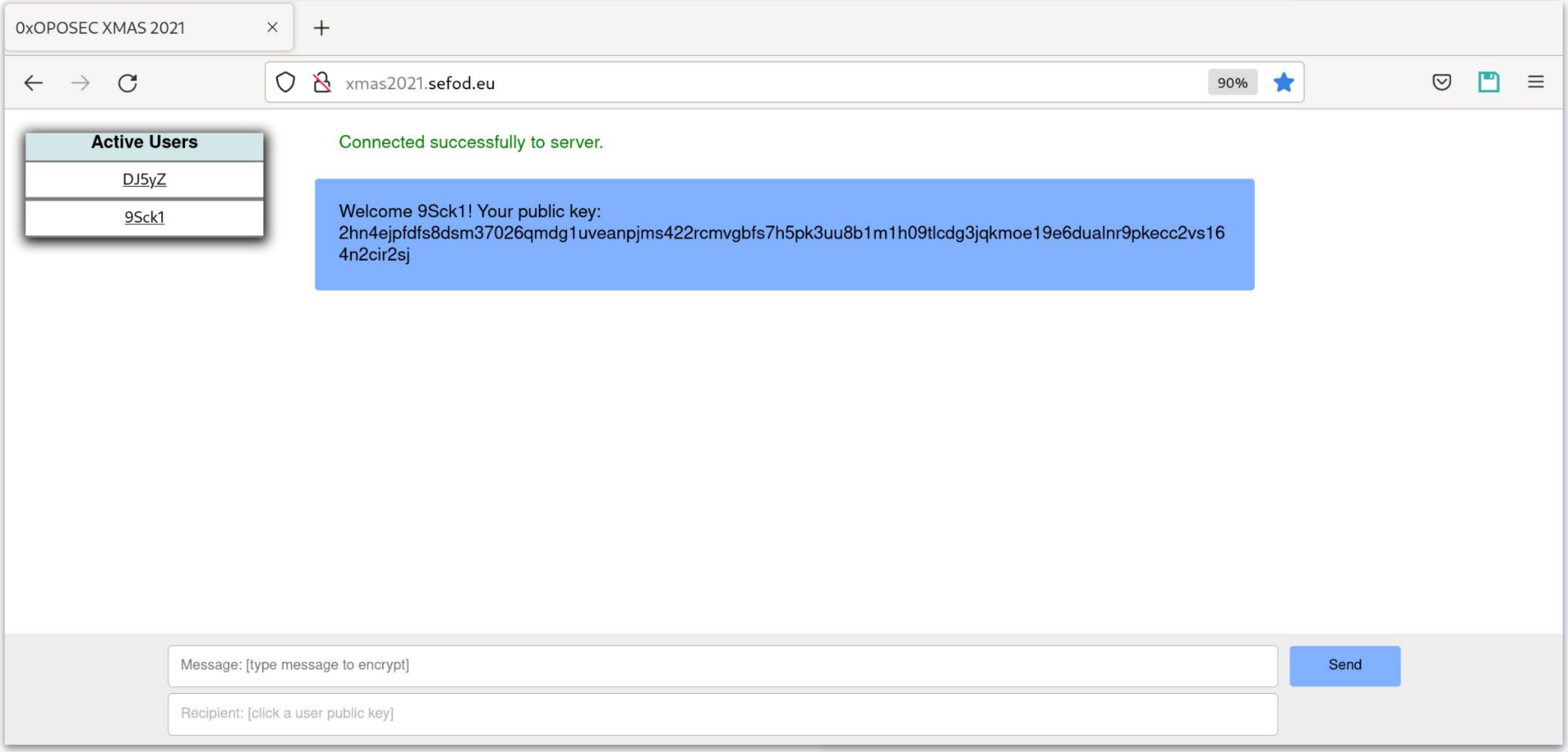
XMAS Challenge 2021

Write-Up

http://xmas2021.sefod.eu/



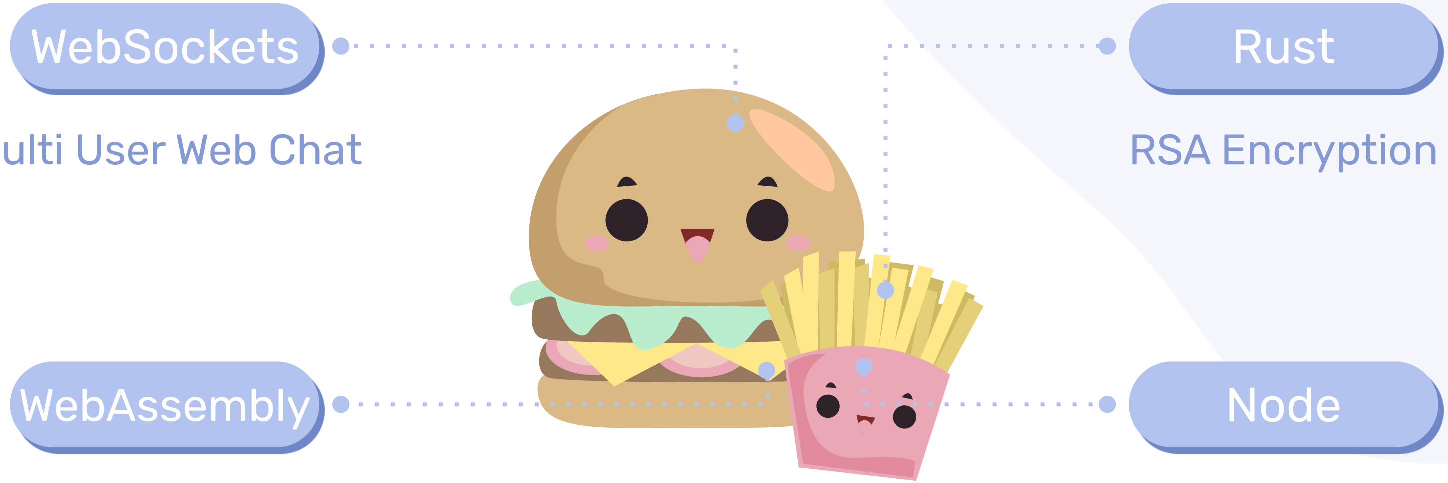
Connected successfully to server.

Welcome 9Sck1! Your public key:
2hn4ejpfdfs8dsm37026qmdg1uveanpjms422rcmvgbfs7h5pk3uu8b1m1h09tlcdg3jqkmoe19e6dualnr9pkecc2vs164n2cir2sj

Message: [type message to encrypt]

Recipient: [click a user public key]

Infographics may be useful



WebSockets

Multi User Web Chat

Rust

RSA Encryption

WebAssembly

WASM Challenge
RSA Encryption

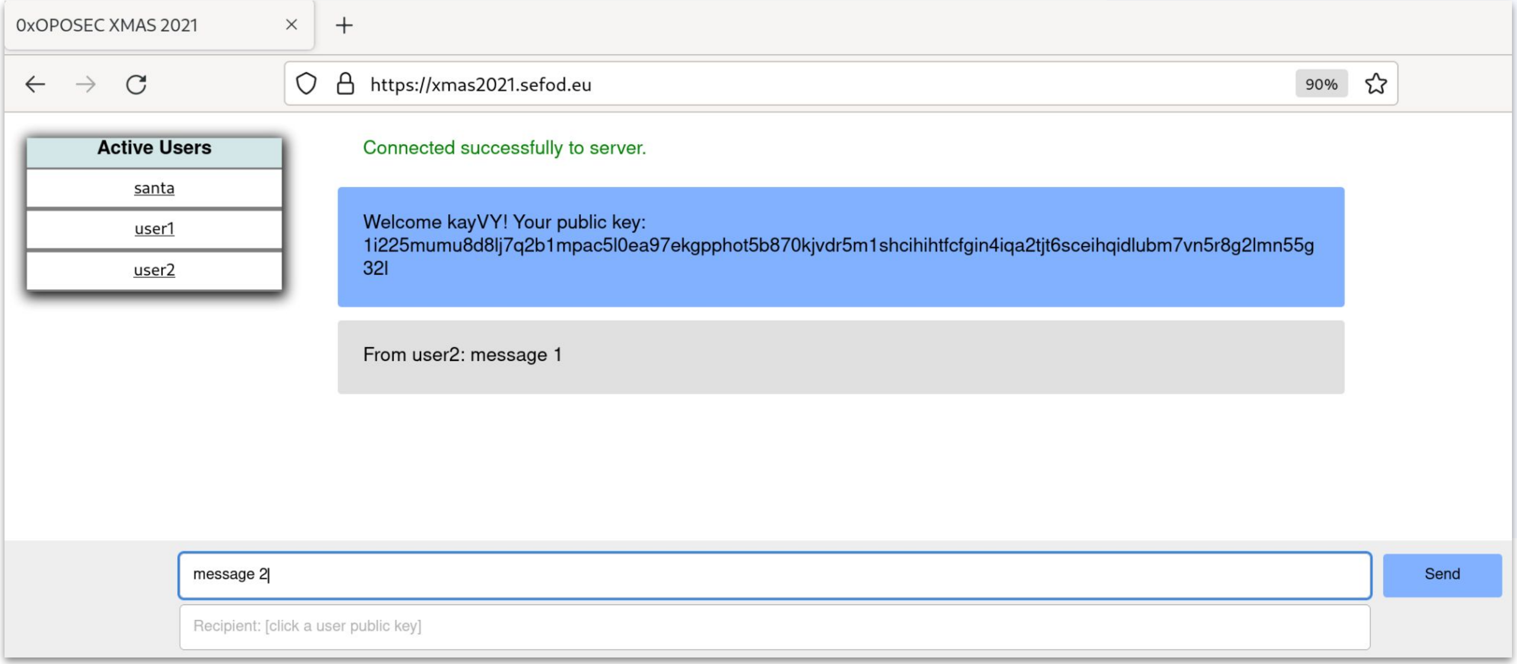
Node

Web Site

01

Basic Features

Broadcast Messages



Active Users

santa
user1
user2

Connected successfully to server.

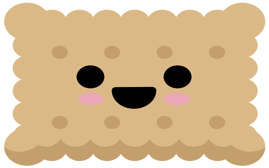
Welcome kayVY! Your public key:
1i225mumu8d8lj7q2b1mpac5l0ea97ekgpphot5b870kjvdr5m1shcihihtfcfgin4iqa2tjt6sceihqidlubm7vn5r8g2lmn55g32l

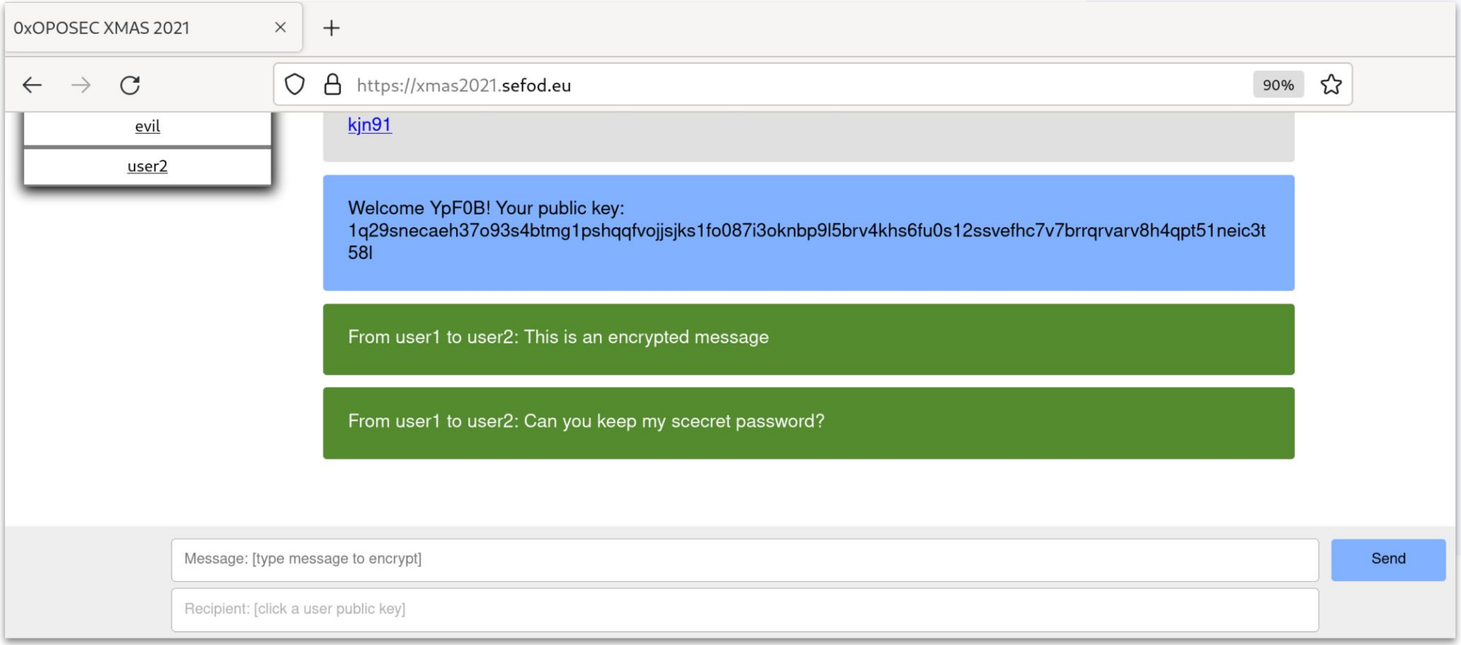
From user2: message 1

message 2

Recipient: [click a user public key]

Private Messages Through Encryption





Welcome YpF0B! Your public key: 1q29snecaeh37o93s4btmg1pshqqfvojjsjks1fo087i3oknbp9l5brv4khs6fu0s12ssvefhc7v7brrqrvarv8h4qpt51neic3t58l

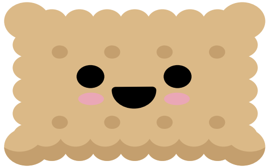
From user1 to user2: This is an encrypted message

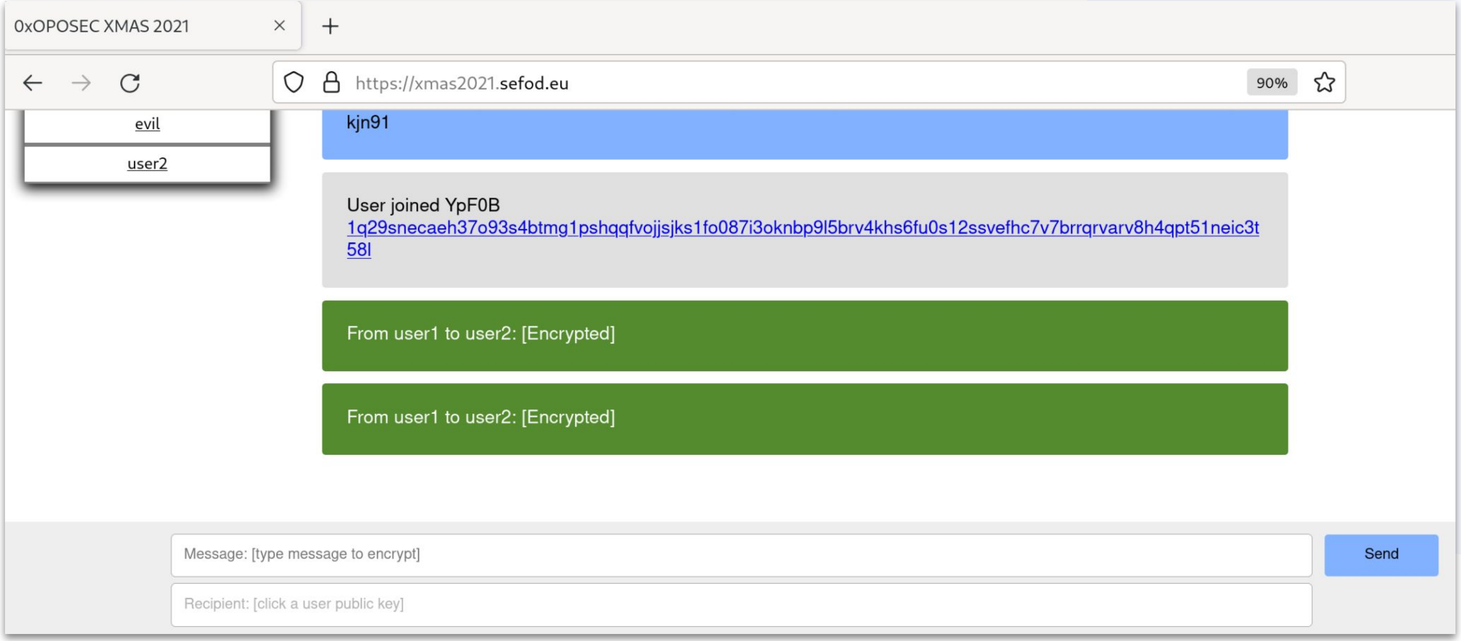
From user1 to user2: Can you keep my scecret password?

Message: [type message to encrypt]

Recipient: [click a user public key]

Private Messages Through Encryption





kjn91

User joined YpF0B
1q29snecaeh37o93s4btmg1pshqqfvojjsjks1fo087i3oknbp9l5brv4khs6fu0s12ssvefhc7v7brrqrvarv8h4qpt51neic3t58l

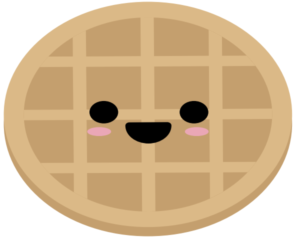
From user1 to user2: [Encrypted]

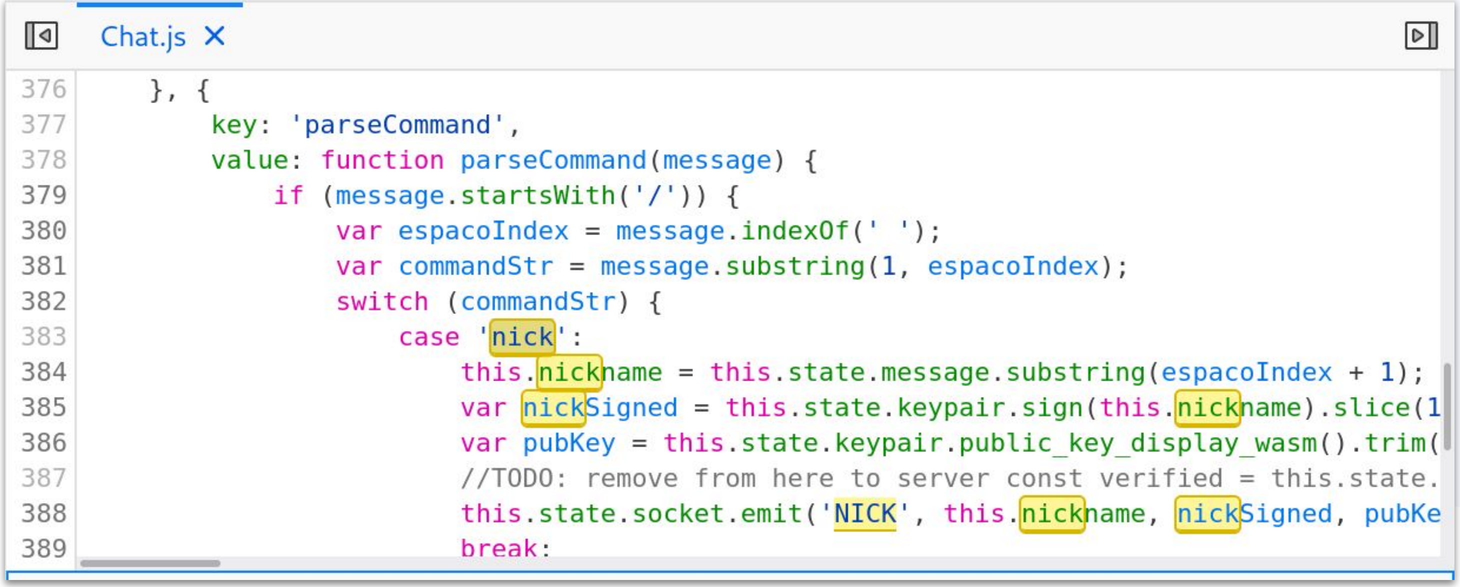
From user1 to user2: [Encrypted]

Message: [type message to encrypt]

Recipient: [click a user public key]

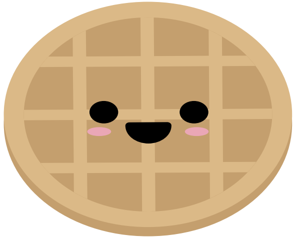
Change Nick Command

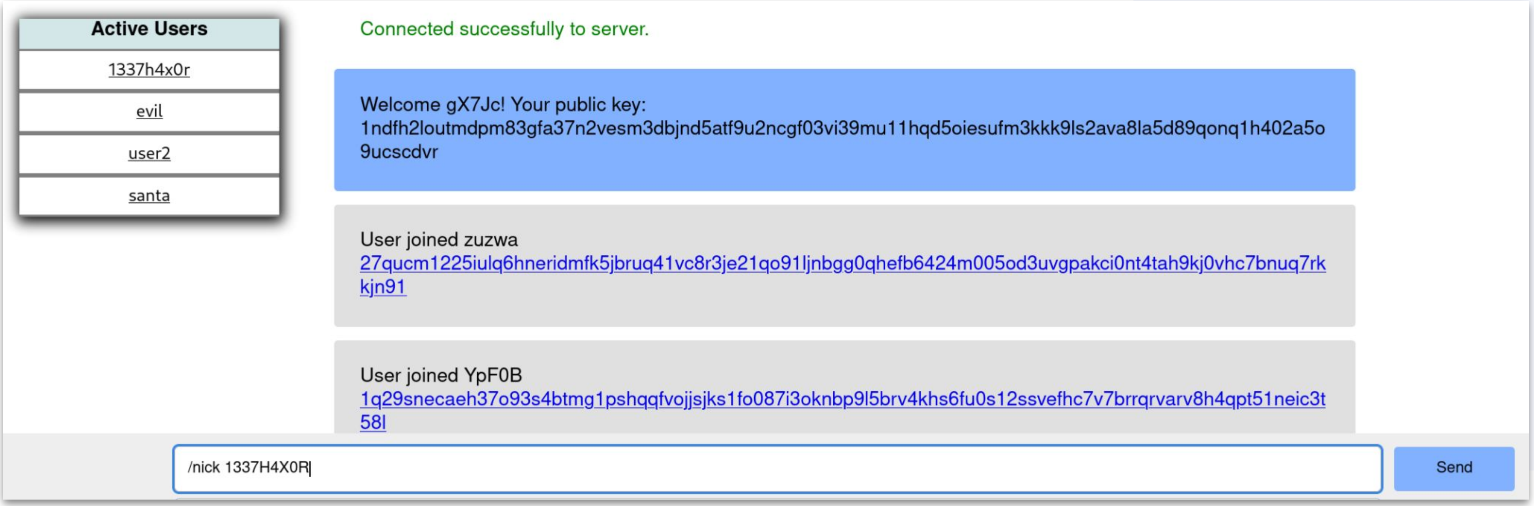




```
    }, {
        key: 'parseCommand',
        value: function parseCommand(message) {
            if (message.startsWith('/')) {
                var espacoIndex = message.indexOf(' ');
                var commandStr = message.substring(1, espacoIndex);
                switch (commandStr) {
                    case 'nick':
                        this.nickname = this.state.message.substring(espacoIndex + 1);
                        var nickSigned = this.state.keypair.sign(this.nickname).slice(1
                        var pubKey = this.state.keypair.public_key_display_wasm().trim(
                        //TODO: remove from here to server const verified = this.state.
                        this.state.socket.emit('NICK', this.nickname, nickSigned, pubKe
                        break;
```

Change Nick Command





Connected successfully to server.

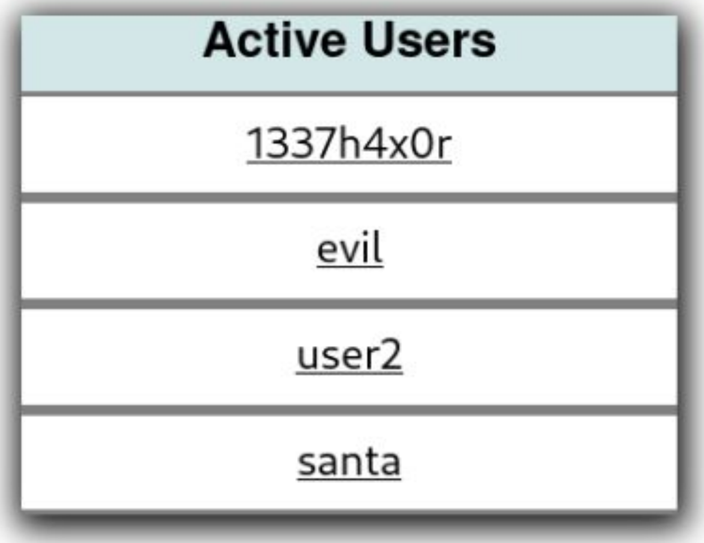
Welcome gX7Jc! Your public key:
1ndfh2loutmdpm83gfa37n2vesm3dbjnd5atf9u2ncgf03vi39mu11hqd5oiesufm3kkk9ls2ava8la5d89qonq1h402a5o9ucscdvr

User joined zuzwa
27qucm1225iulq6hneridmfk5jbruq41vc8r3je21qo91ljnbgg0qhefb6424m005od3uvgpakci0nt4tah9kj0vhc7bnuq7rkkjn91

User joined YpF0B
1q29snecaeh37o93s4btmg1pshqqfvojjsjks1fo087i3oknbp9l5brv4khs6fu0s12ssvefhc7v7brrqrvarv8h4qpt51neic3t58l

```
/nick 1337H4X0R
```

Bot Named Santa

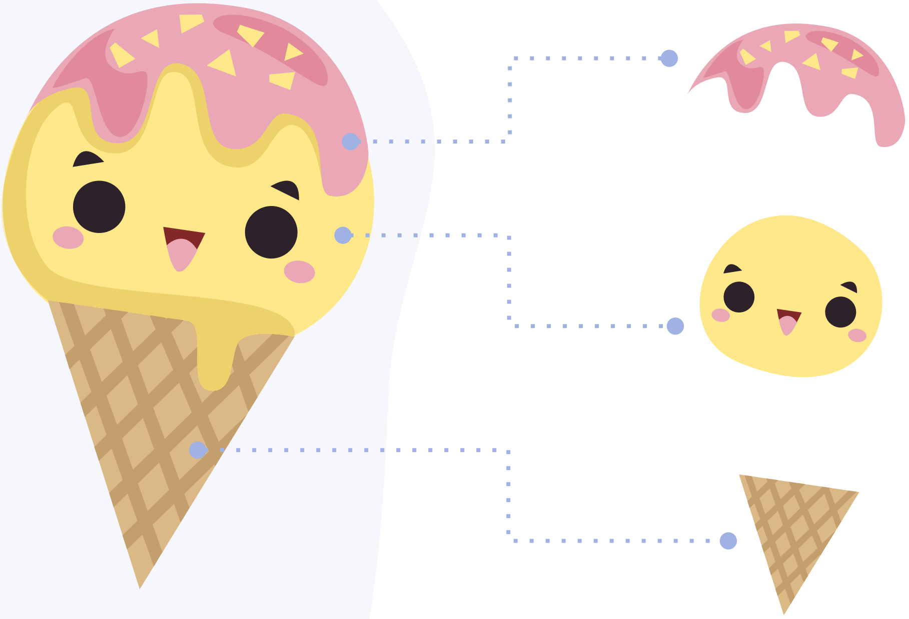


Active Users
1337h4x0r
evil
user2
santa

3 Flags

Reverse The Chat and Discover the Flags

Flags



WebSocket Flag

Santa JavaScript Flag

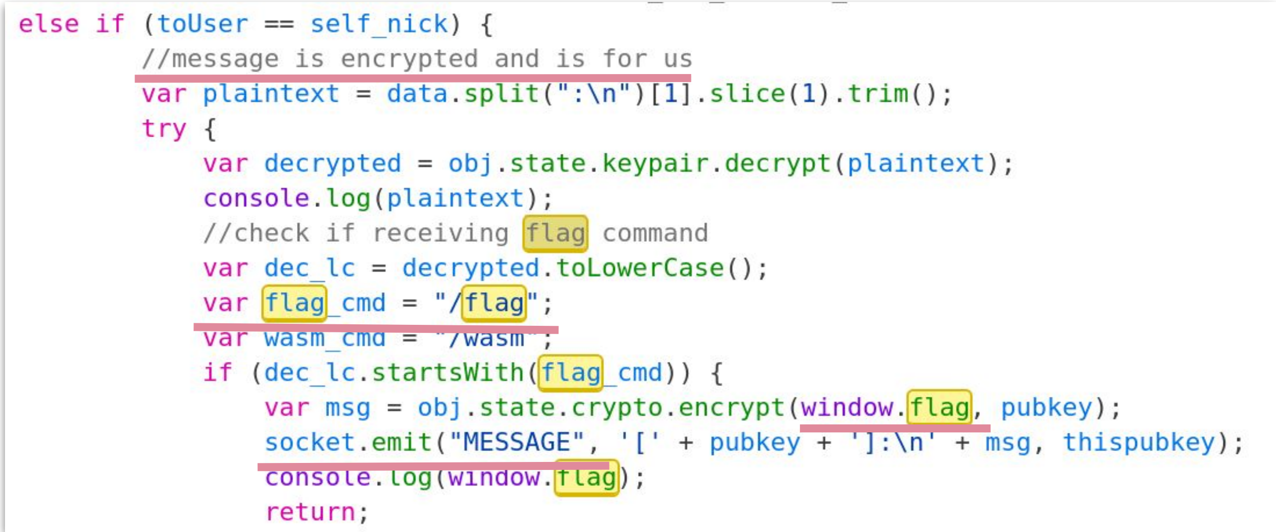
Santa Wasm Flag

02

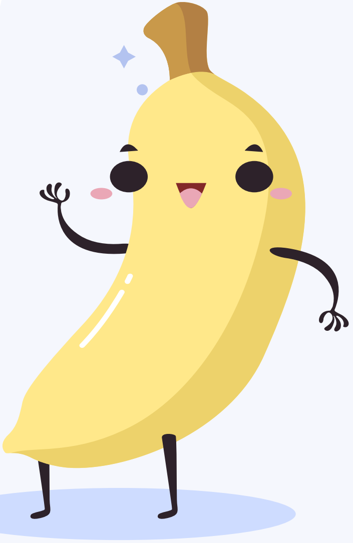
Reversing WebSite

JavaScript Is Not Obfuscated

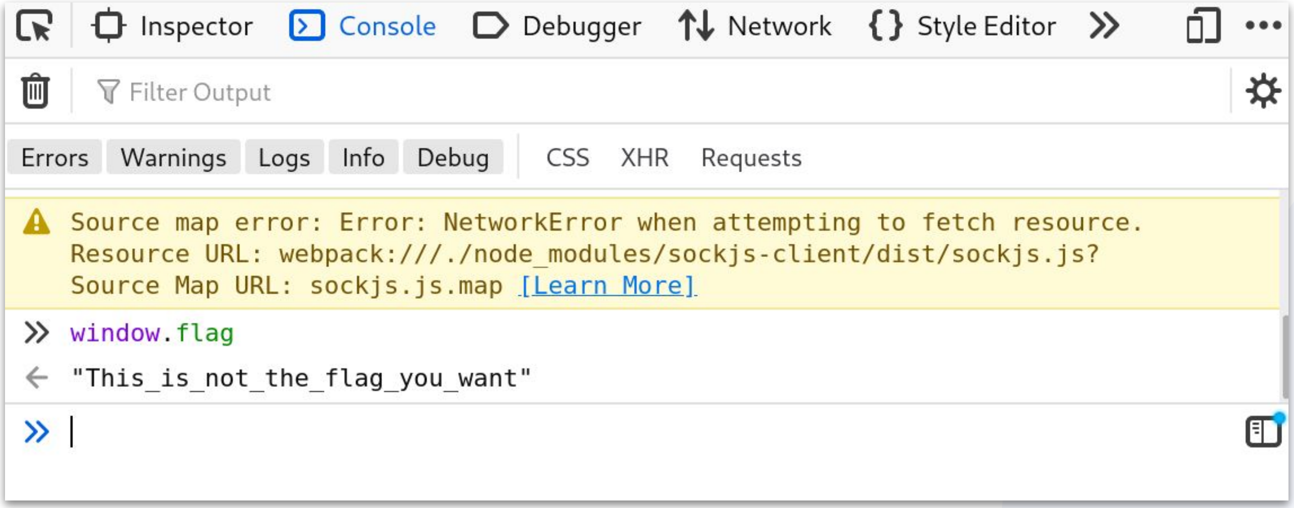
/Flag Command



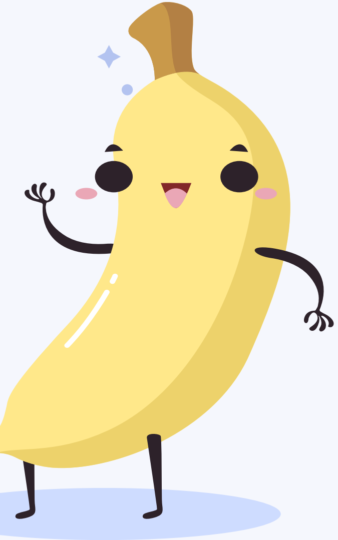
```
else if (toUser == self_nick) {
        //message is encrypted and is for us
        var plaintext = data.split(":\n")[1].slice(1).trim();
        try {
            var decrypted = obj.state.keypair.decrypt(plaintext);
            console.log(plaintext);
            //check if receiving flag command
            var dec_lc = decrypted.toLowerCase();
            var flag_cmd = "/flag";
            var wasm_cmd = "/wasm";
            if (dec_lc.startsWith(flag_cmd)) {
                var msg = obj.state.crypto.encrypt(window.flag, pubkey);
                socket.emit("MESSAGE", '[' + pubkey + ']:\n' + msg, thispubkey);
                console.log(window.flag);
                return;
```



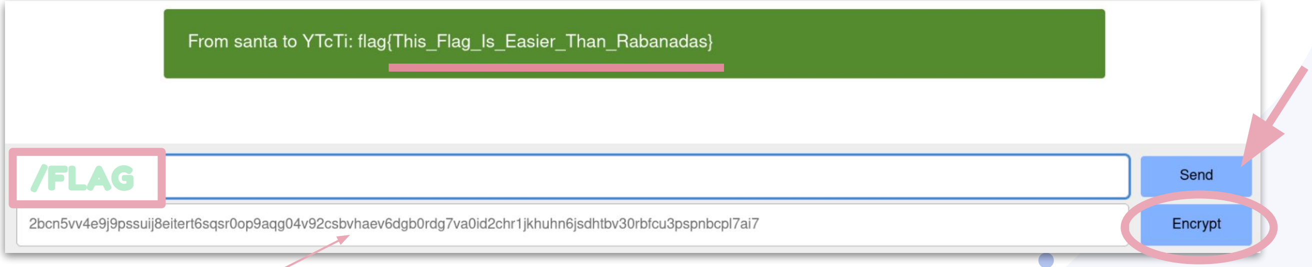
/Flag Command



```
⚠ Source map error: Error: NetworkError when attempting to fetch resource.
  Resource URL: webpack:///./node_modules/sockjs-client/dist/sockjs.js?
  Source Map URL: sockjs.js.map [Learn More]
>> window.flag
← "This_is_not_the_flag_you_want"
>>
```



/Flag Command



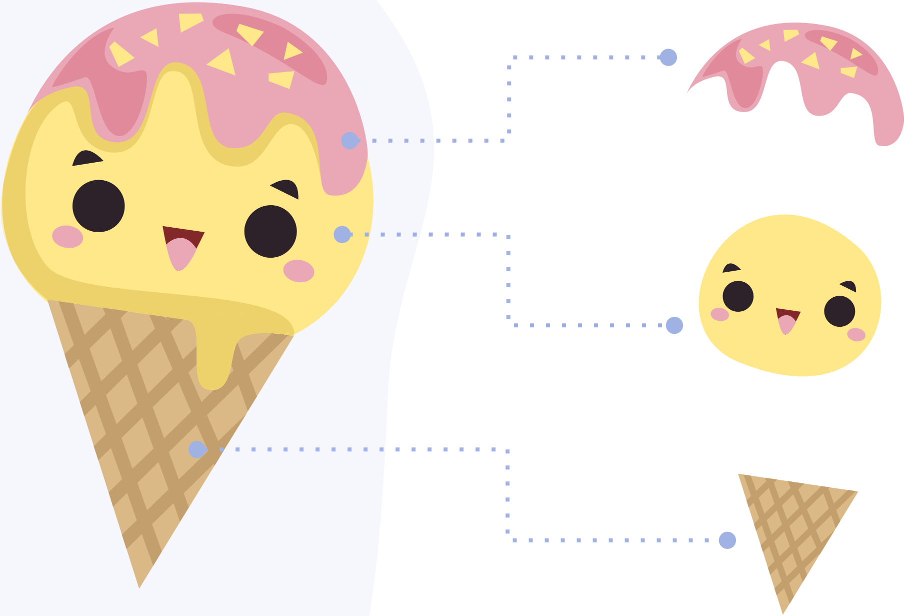
From santa to YTcTi: flag{This_Flag_Is_Easier_Than_Rabanadas}

/FLAG

2bcn5vv4e9j9pssuij8eitert6sqsr0op9aqg04v92csbvhaev6dgb0rdg7va0id2chr1jkhuhn6jsdhtbv30rbfcu3pspnbcpl7ai7

Santa PubKey

Flags



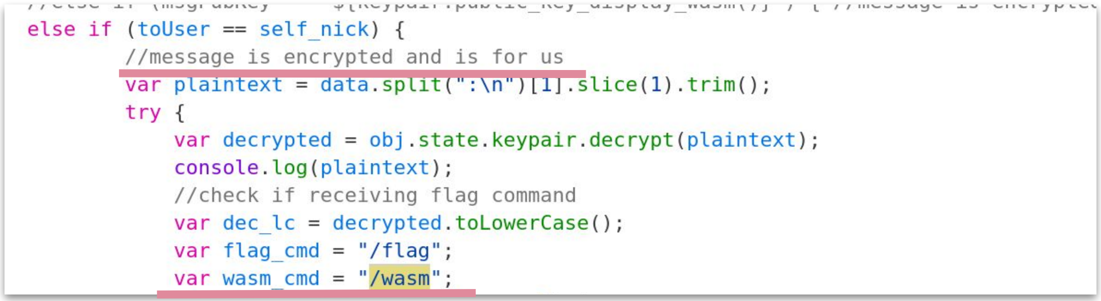
WebSocket Flag

Santa JavaScript Flag

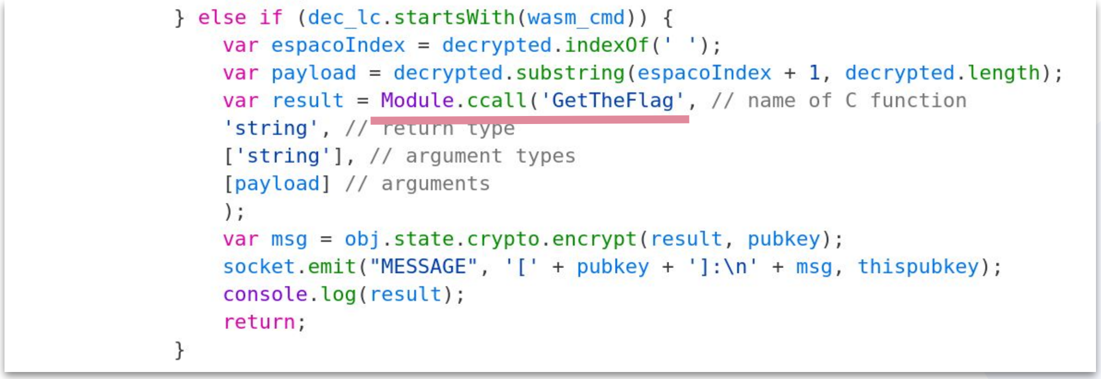
flag{This_Flag_Is_Easier_Than_Rabanadas}

Santa Wasm Flag

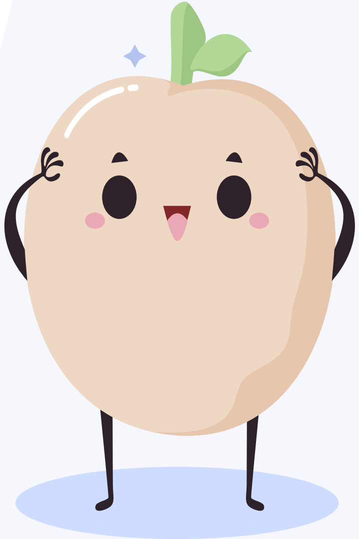
/Wasm Command



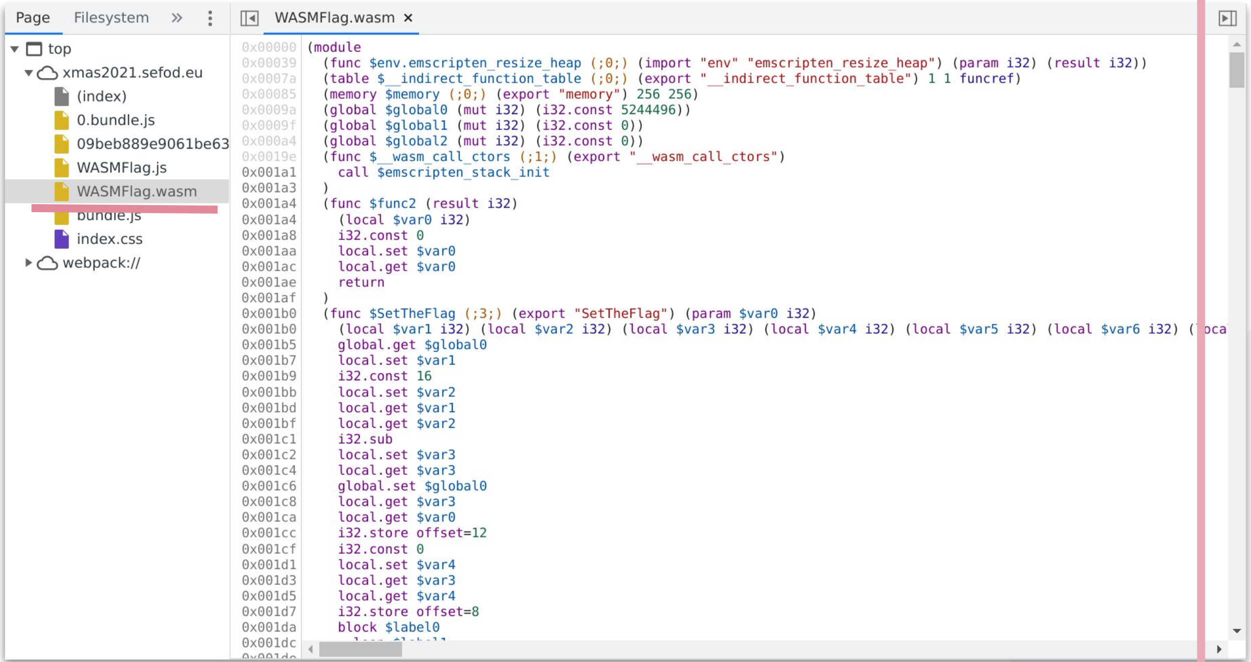
```
//else if (msgPubKey == ${keypair.public_key_display_wasm()} ) { //message is encrypted
else if (toUser == self_nick) {
        //message is encrypted and is for us
        var plaintext = data.split(":\n")[1].slice(1).trim();
        try {
            var decrypted = obj.state.keypair.decrypt(plaintext);
            console.log(plaintext);
            //check if receiving flag command
            var dec_lc = decrypted.toLowerCase();
            var flag_cmd = "/flag";
            var wasm_cmd = "/wasm";
```



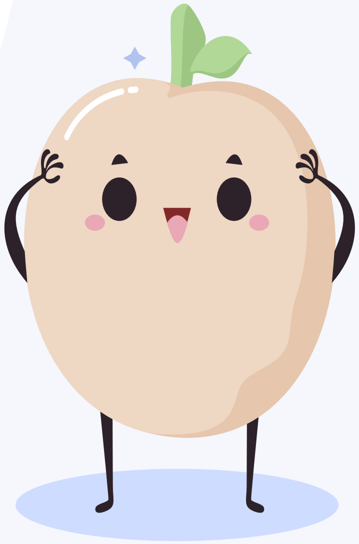
```
            } else if (dec_lc.startsWith(wasm_cmd)) {
                var espacoIndex = decrypted.indexOf(' ');
                var payload = decrypted.substring(espacoIndex + 1, decrypted.length);
                var result = Module.ccall('GetTheFlag', // name of C function
                'string', // return type
                ['string'], // argument types
                [payload] // arguments
                );
                var msg = obj.state.crypto.encrypt(result, pubkey);
                socket.emit("MESSAGE", '[' + pubkey + ']:\n' + msg, thispubkey);
                console.log(result);
                return;
            }
```



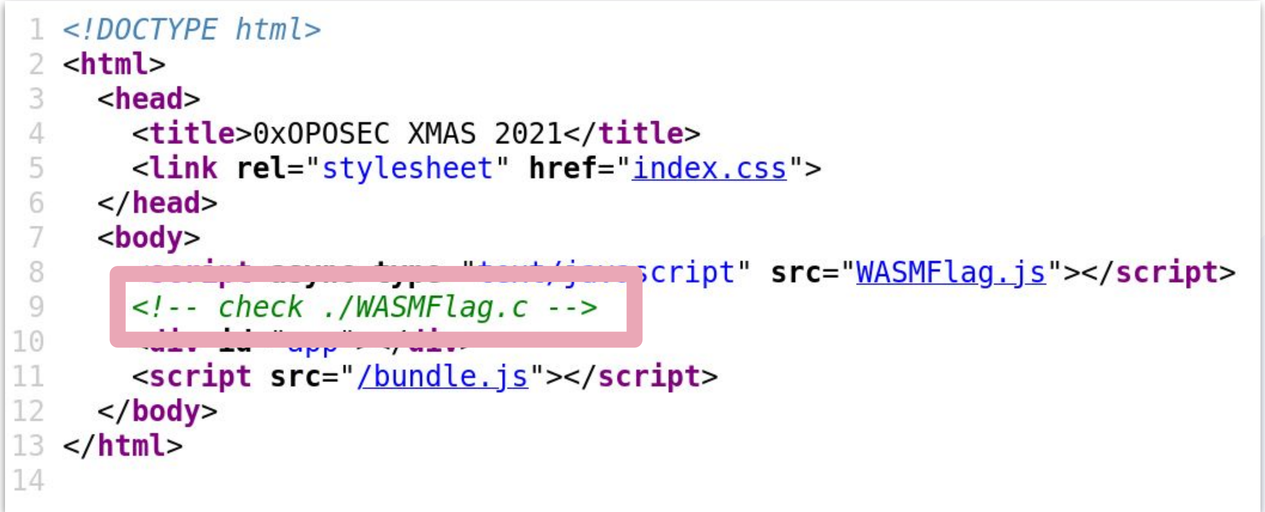
/Wasm Command



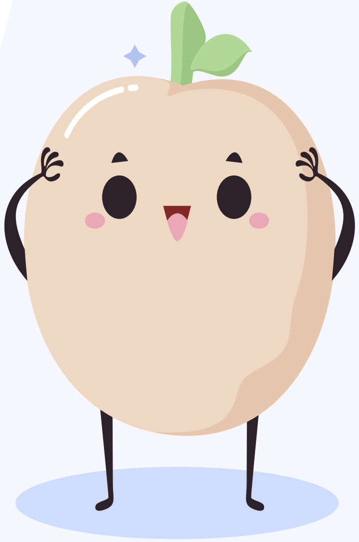
```
0x00000 (module
0x00039   (func $env.emscripten_resize_heap (;0;) (import "env" "emscripten_resize_heap") (param i32) (result i32))
0x0007a   (table $__indirect_function_table (;0;) (export "__indirect_function_table") 1 1 funcref)
0x00085   (memory $memory (;0;) (export "memory") 256 256)
0x0009a   (global $global0 (mut i32) (i32.const 5244496))
0x0009f   (global $global1 (mut i32) (i32.const 0))
0x000a4   (global $global2 (mut i32) (i32.const 0))
0x0019e   (func $__wasm_call_ctors (;1;) (export "__wasm_call_ctors")
0x001a1     call $emscripten_stack_init
0x001a3   )
0x001a4   (func $func2 (result i32)
0x001a4     (local $var0 i32)
0x001a8     i32.const 0
0x001aa     local.set $var0
0x001ac     local.get $var0
0x001ae     return
0x001af   )
0x001b0   (func $SetTheFlag (;3;) (export "SetTheFlag") (param $var0 i32)
0x001b0     (local $var1 i32) (local $var2 i32) (local $var3 i32) (local $var4 i32) (local $var5 i32) (local $var6 i32) (loca
0x001b5     global.get $global0
0x001b7     local.set $var1
0x001b9     i32.const 16
0x001bb     local.set $var2
0x001bd     local.get $var1
0x001bf     local.get $var2
0x001c1     i32.sub
0x001c2     local.set $var3
0x001c4     local.get $var3
0x001c6     global.set $global0
0x001c8     local.get $var3
0x001ca     local.get $var0
0x001cc     i32.store offset=12
0x001cf     i32.const 0
0x001d1     local.set $var4
0x001d3     local.get $var3
0x001d5     local.get $var4
0x001d7     i32.store offset=8
0x001da     block $label0
0x001dc       loop $label1
0x001de
```



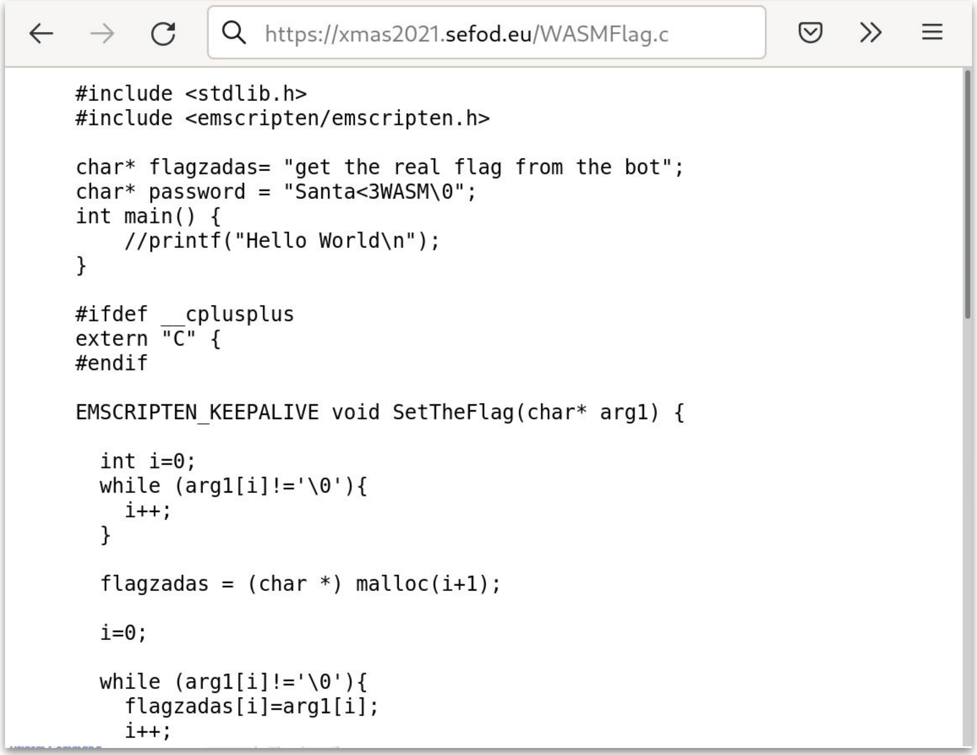
/Wasm Command



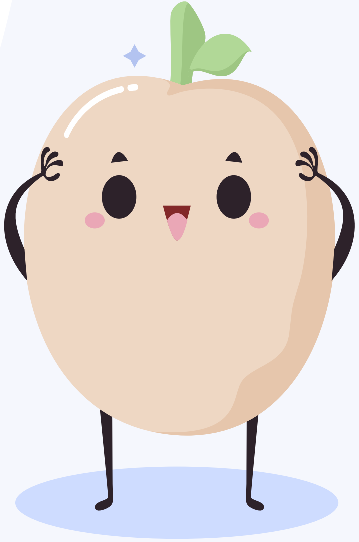
```
<!DOCTYPE html>
<html>
  <head>
    <title>0xOPOSEC XMAS 2021</title>
    <link rel="stylesheet" href="index.css">
  </head>
  <body>
    <script async type="text/javascript" src="WASMFlag.js"></script>
    <!-- check ./WASMFlag.c -->
    <div id="app"></div>
    <script src="/bundle.js"></script>
  </body>
</html>
```



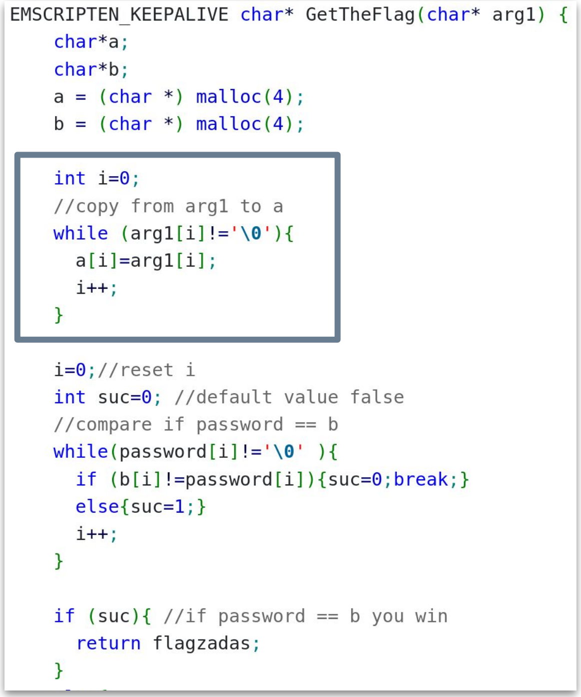
/Wasm Command



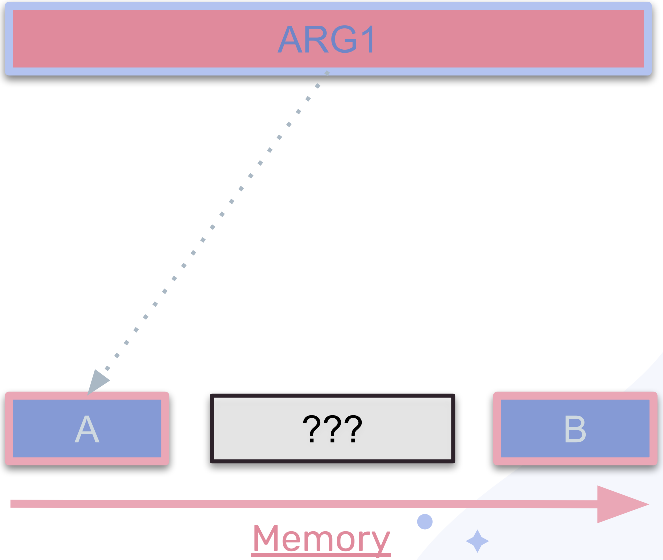
```
#include <stdlib.h>
#include <emscripten/emscripten.h>

char* flagzadas= "get the real flag from the bot";
char* password = "Santa<3WASM\0";
int main() {
    //printf("Hello World\n");
}

#ifdef __cplusplus
extern "C" {
#endif

EMSCRIPTEN_KEEPALIVE void SetTheFlag(char* arg1) {

  int i=0;
  while (arg1[i]!='\0'){
    i++;
  }

  flagzadas = (char *) malloc(i+1);

  i=0;

  while (arg1[i]!='\0'){
    flagzadas[i]=arg1[i];
    i++;
```

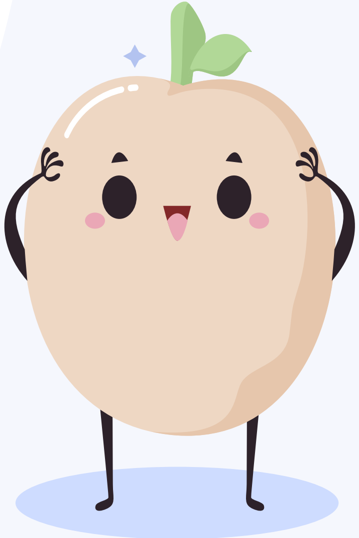


/Wasm Command

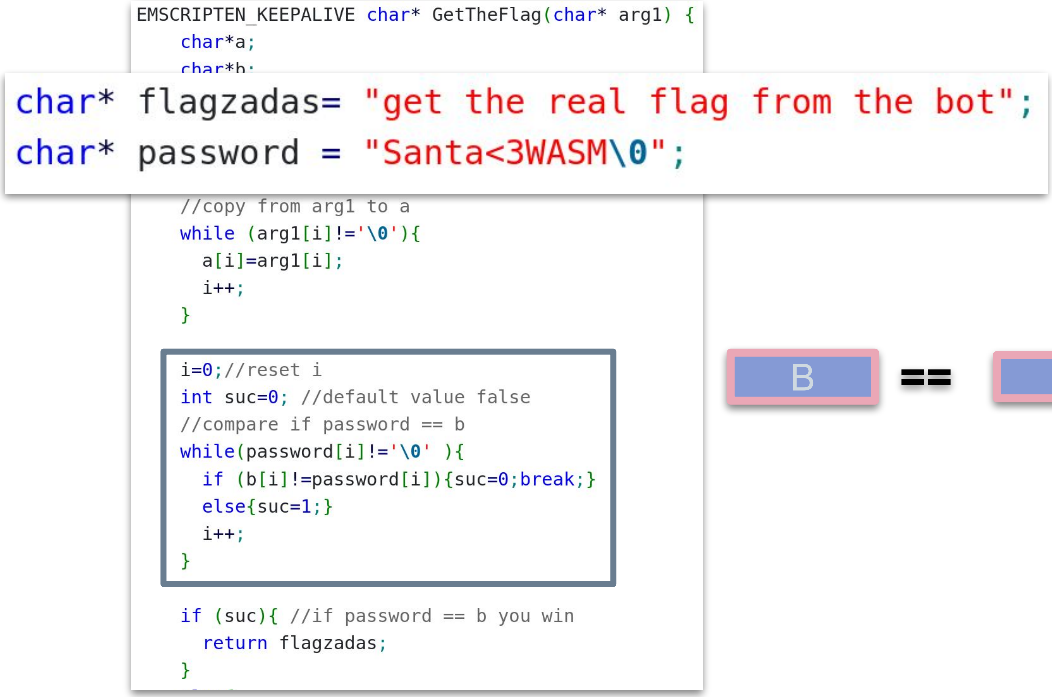


```
EMSCRIPTEN_KEEPALIVE char* GetTheFlag(char* arg1) {
    char*a;
    char*b;
    a = (char *) malloc(4);
    b = (char *) malloc(4);

    int i=0;
    //copy from arg1 to a
    while (arg1[i]!='\0'){
      a[i]=arg1[i];
      i++;
    }

    i=0;//reset i
    int suc=0; //default value false
    //compare if password == b
    while(password[i]!='\0' ){
      if (b[i]!=password[i]){suc=0;break;}
      else{suc=1;}
      i++;
    }

    if (suc){ //if password == b you win
      return flagzadas;
    }
```





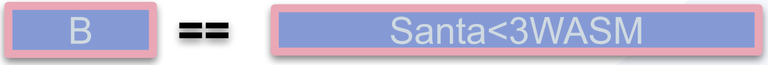
/Wasm Command



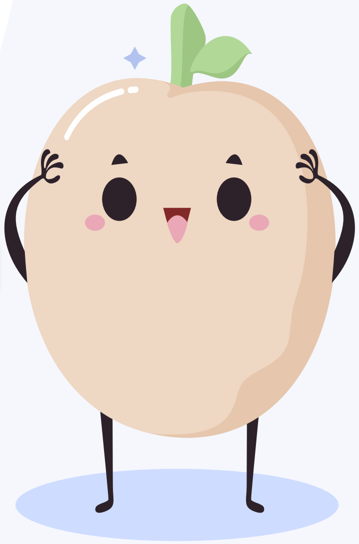
```
EMSCRIPTEN_KEEPALIVE char* GetTheFlag(char* arg1) {
    char*a;
    char*b;
```

```
char* flagzadas= "get the real flag from the bot";
char* password = "Santa<3WASM\0";
```

```
    //copy from arg1 to a
    while (arg1[i]!='\0'){
      a[i]=arg1[i];
      i++;
    }

    i=0;//reset i
    int suc=0; //default value false
    //compare if password == b
    while(password[i]!='\0' ){
      if (b[i]!=password[i]){suc=0;break;}
      else{suc=1;}
      i++;
    }

    if (suc){ //if password == b you win
      return flagzadas;
    }
```

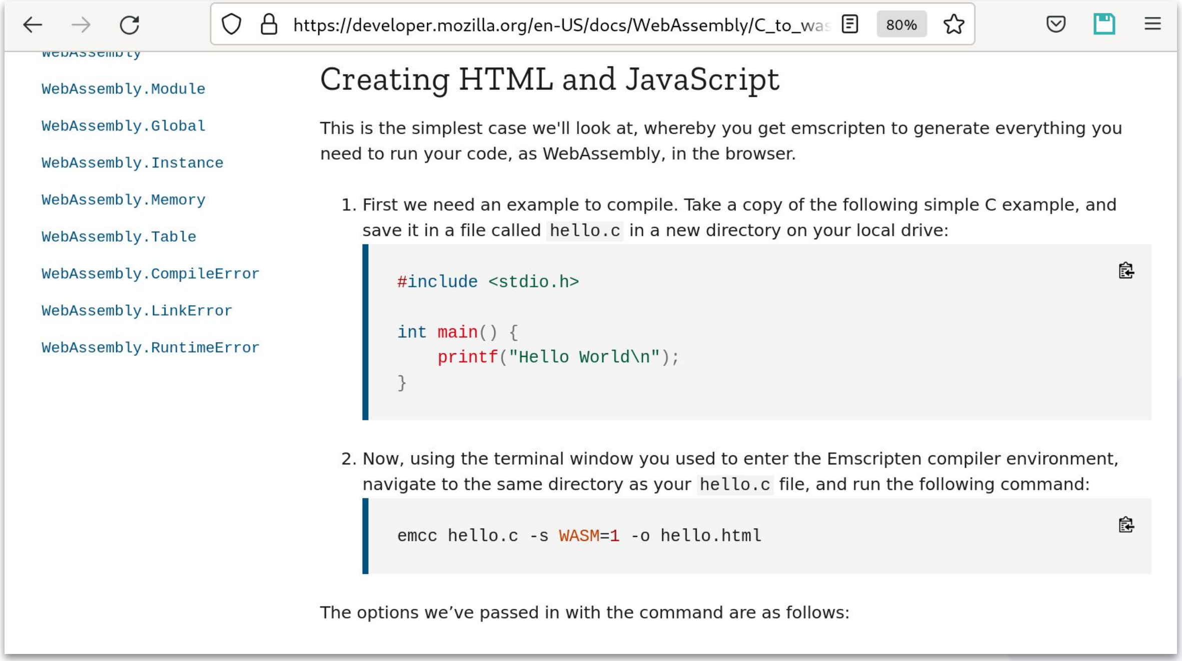


B == Santa<3WASM



Testing Wasm Offline

Testing Wasm



https://developer.mozilla.org/en-US/docs/WebAssembly/C_to_was

WebAssembly.Module

WebAssembly.Global

WebAssembly.Instance

WebAssembly.Memory

WebAssembly.Table

WebAssembly.CompileError

WebAssembly.LinkError

WebAssembly.RuntimeError

Creating HTML and JavaScript

This is the simplest case we'll look at, whereby you get emscripten to generate everything you need to run your code, as WebAssembly, in the browser.

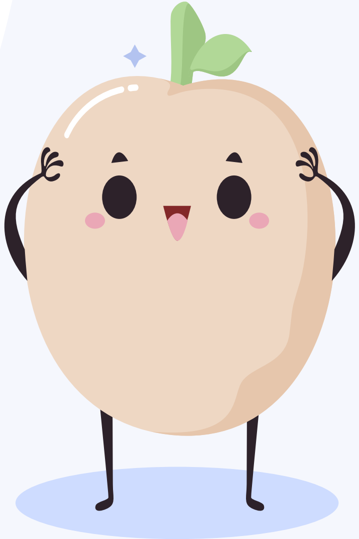
1. First we need an example to compile. Take a copy of the following simple C example, and save it in a file called `hello.c` in a new directory on your local drive:

```
#include <stdio.h>

int main() {
    printf("Hello World\n");
}
```

2. Now, using the terminal window you used to enter the Emscripten compiler environment, navigate to the same directory as your `hello.c` file, and run the following command:

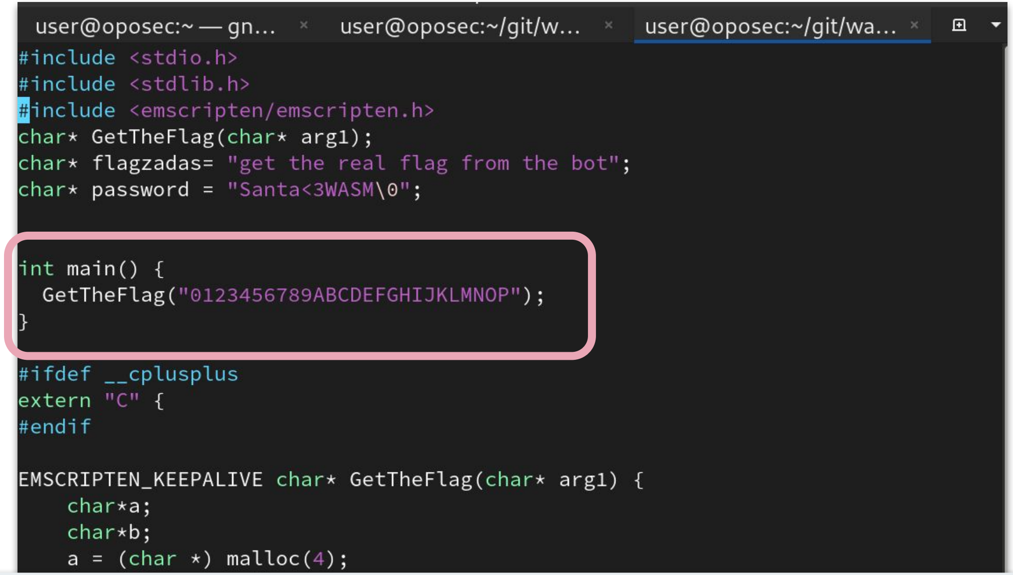
```
emcc hello.c -s WASM=1 -o hello.html
```

The options we've passed in with the command are as follows:

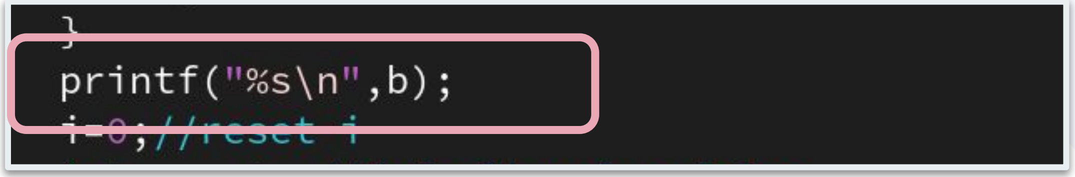


https://developer.mozilla.org/en-US/docs/WebAssembly/C_to_wasm

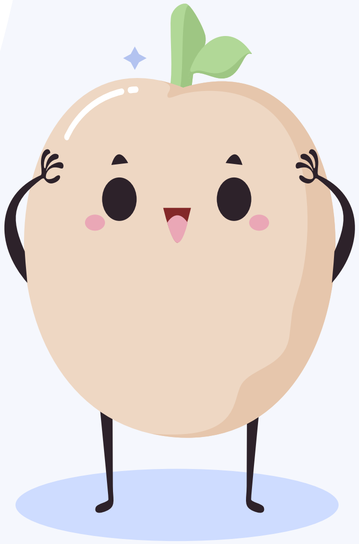
Testing Wasm



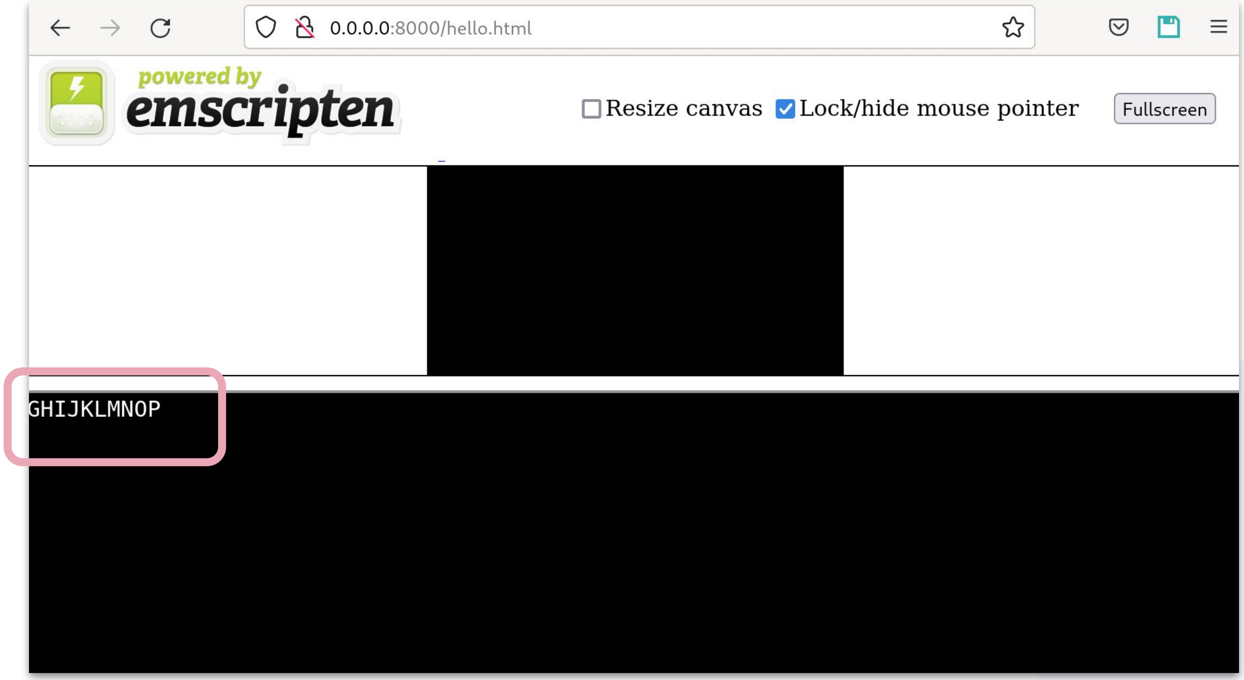
```
#include <stdio.h>
#include <stdlib.h>
#include <emscripten/emscripten.h>
char* GetTheFlag(char* arg1);
char* flagzadas= "get the real flag from the bot";
char* password = "Santa<3WASM\0";

int main() {
  GetTheFlag("0123456789ABCDEFGHIJKLMNOP");
}

#ifdef __cplusplus
extern "C" {
#endif

EMSCRIPTEN_KEEPALIVE char* GetTheFlag(char* arg1) {
    char*a;
    char*b;
    a = (char *) malloc(4);
```



```
  }
  printf("%s\n",b);
  i=0;//reset i
```

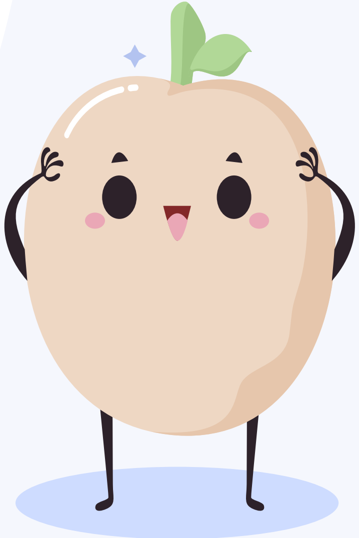


Testing Wasm

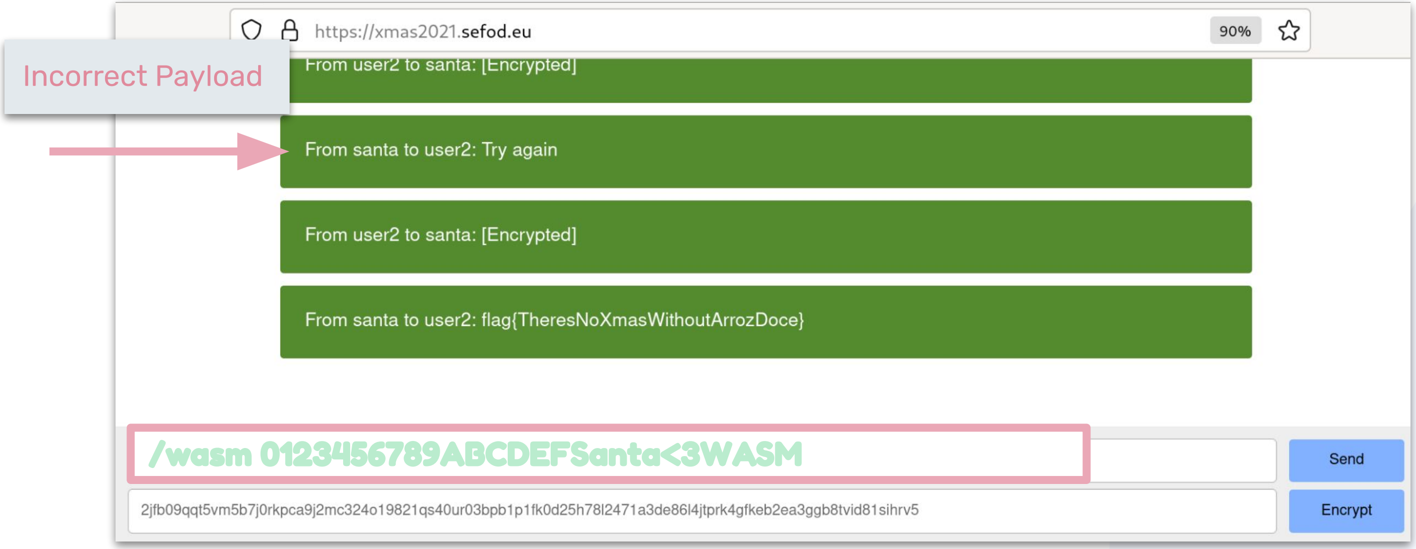


☐ Resize canvas ☑ Lock/hide mouse pointer

```
GHIJKLMNOP
```



Exploiting Wasm



Incorrect Payload

From user2 to santa: [Encrypted]

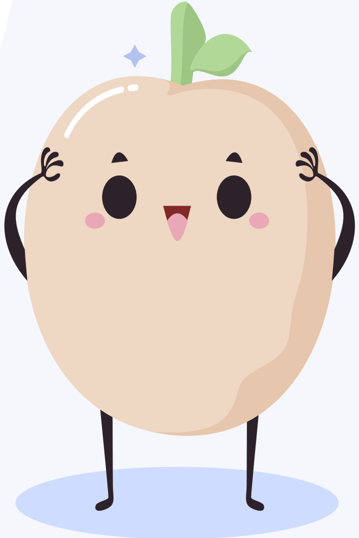
From santa to user2: Try again

From user2 to santa: [Encrypted]

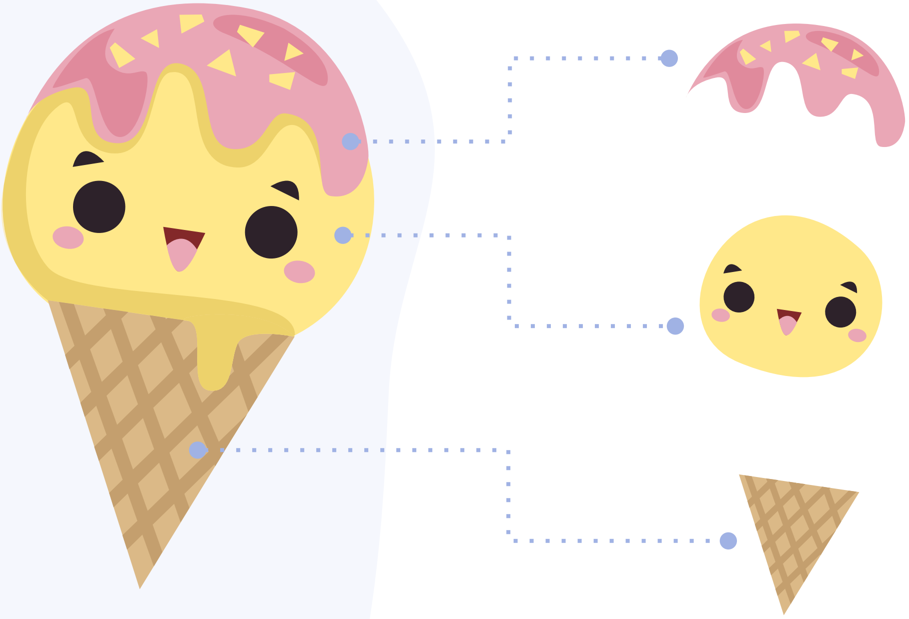
From santa to user2: flag{TheresNoXmasWithoutArrozDoce}

/wasm 0123456789ABCDEFSanta<3WASM

2jfb09qqt5vm5b7j0rkpca9j2mc324o19821qs40ur03bpb1p1fk0d25h78l2471a3de86l4jtprk4gfkeb2ea3ggb8tvid81sihrv5



Flags



WebSocket Flag

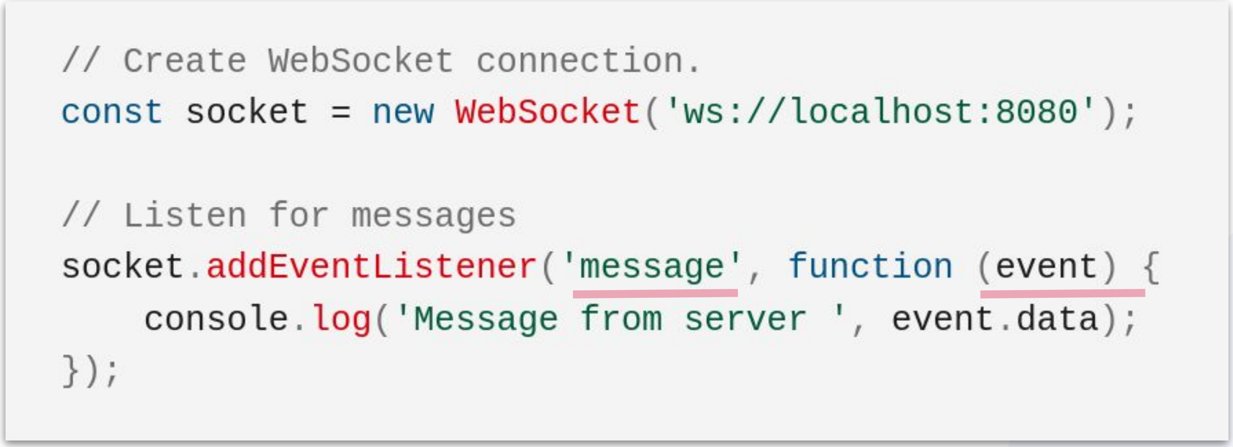
Santa JavaScript Flag

flag{This_Flag_Is_Easier_Than_Rabanadas}

Santa Wasm Flag

flag{TheresNoXmasWithoutArrozDoce}

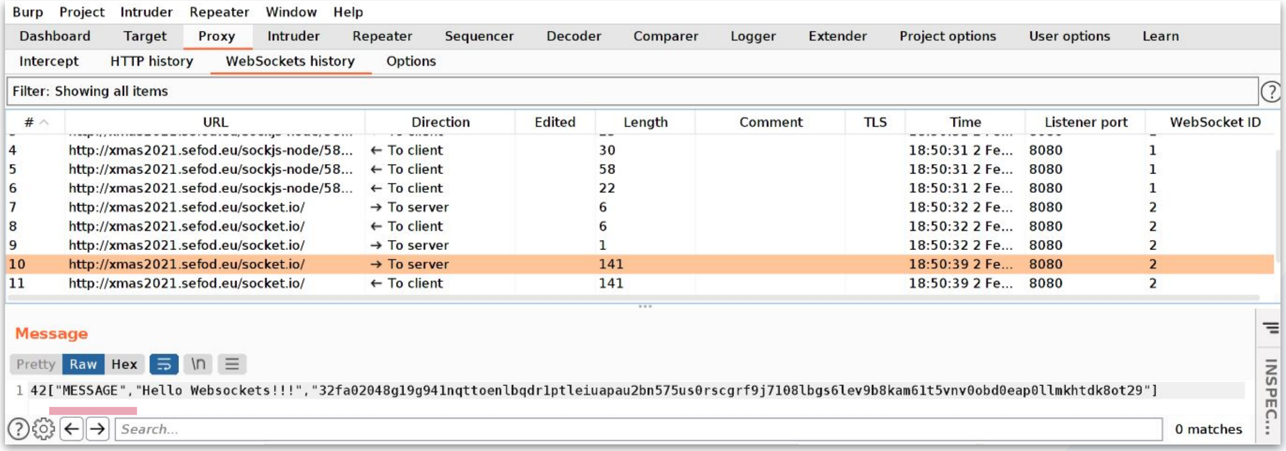
Understanding WebSockets



```
// Create WebSocket connection.
const socket = new WebSocket('ws://localhost:8080');

// Listen for messages
socket.addEventListener('message', function (event) {
    console.log('Message from server ', event.data);
});
```

https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_servers

Exploring WebSockets



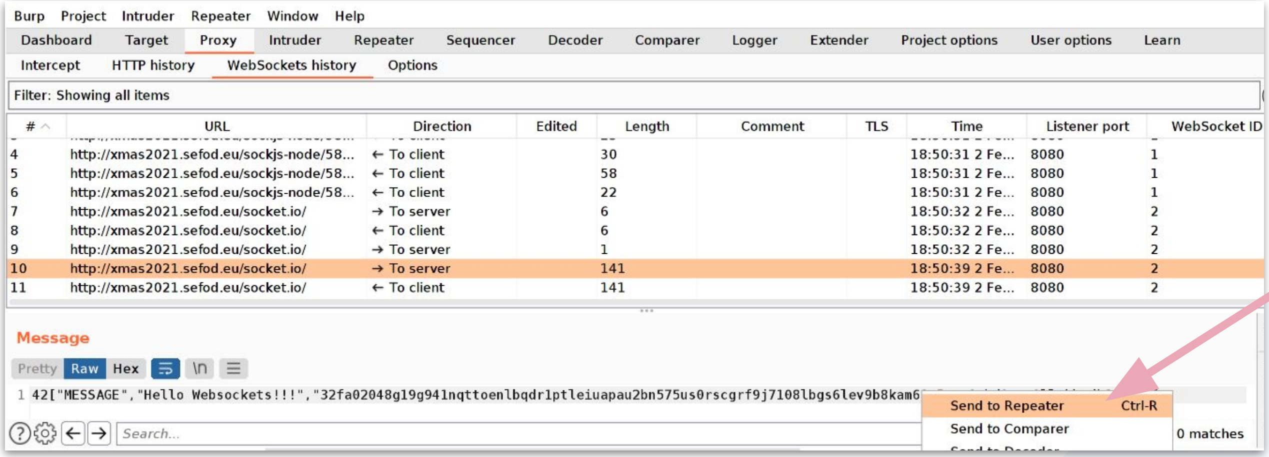
Filter: Showing all items

#	URL	Direction	Edited	Length	Comment	TLS	Time	Listener port	WebSocket ID
4	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		30			18:50:31 2 Fe...	8080	1
5	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		58			18:50:31 2 Fe...	8080	1
6	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		22			18:50:31 2 Fe...	8080	1
7	http://xmas2021.sefod.eu/socket.io/	→ To server		6			18:50:32 2 Fe...	8080	2
8	http://xmas2021.sefod.eu/socket.io/	← To client		6			18:50:32 2 Fe...	8080	2
9	http://xmas2021.sefod.eu/socket.io/	→ To server		1			18:50:32 2 Fe...	8080	2
10	http://xmas2021.sefod.eu/socket.io/	→ To server		141			18:50:39 2 Fe...	8080	2
11	http://xmas2021.sefod.eu/socket.io/	← To client		141			18:50:39 2 Fe...	8080	2

Message

```
42["MESSAGE","Hello Websockets!!!","32fa02048g19g941nqttoenlbqdr1ptleiuapau2bn575us0rscgrf9j7108lbgs6lev9b8kam61t5vnv0obd0eap0llmkhtdk8ot29"]
```

Exploring WebSockets



Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options

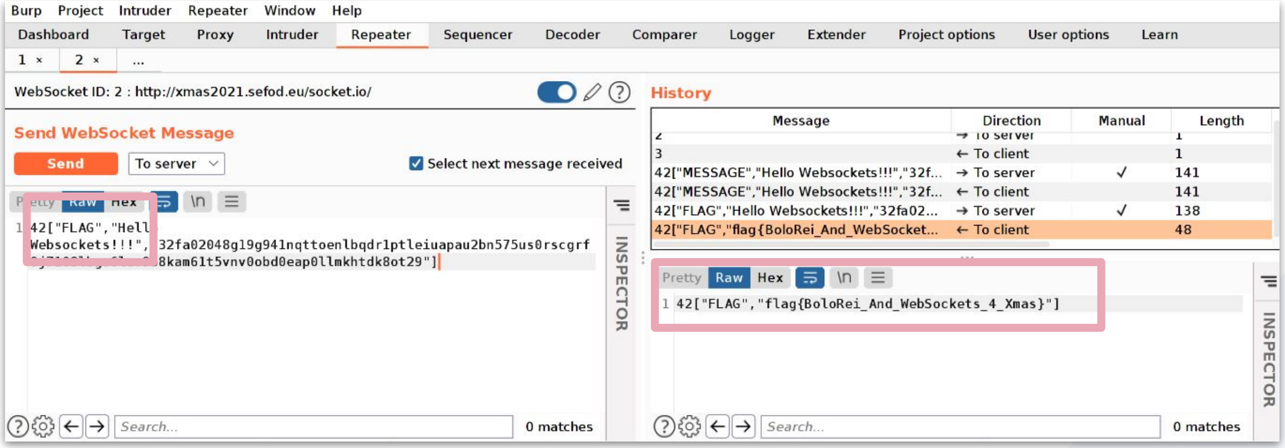
Filter: Showing all items

#	URL	Direction	Edited	Length	Comment	TLS	Time	Listener port	WebSocket ID
4	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		30			18:50:31 2 Fe...	8080	1
5	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		58			18:50:31 2 Fe...	8080	1
6	http://xmas2021.sefod.eu/sockjs-node/58...	← To client		22			18:50:31 2 Fe...	8080	1
7	http://xmas2021.sefod.eu/socket.io/	→ To server		6			18:50:32 2 Fe...	8080	2
8	http://xmas2021.sefod.eu/socket.io/	← To client		6			18:50:32 2 Fe...	8080	2
9	http://xmas2021.sefod.eu/socket.io/	→ To server		1			18:50:32 2 Fe...	8080	2
10	http://xmas2021.sefod.eu/socket.io/	→ To server		141			18:50:39 2 Fe...	8080	2
11	http://xmas2021.sefod.eu/socket.io/	← To client		141			18:50:39 2 Fe...	8080	2

Message

```
42["MESSAGE","Hello Websockets!!!","32fa02048g19g941nqttoenlbqdr1ptleiuapau2bn575us0rscgrf9j7108lbgs6lev9b8kam6
```

Exploring WebSockets


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × 2 × ...
Pretty Raw Hex \n
INSPECTOR
Search...
0 matches

WebSocket ID: 2 : http://xmas2021.sefod.eu/socket.io/

Send WebSocket Message

Send | To server

☑ Select next message received

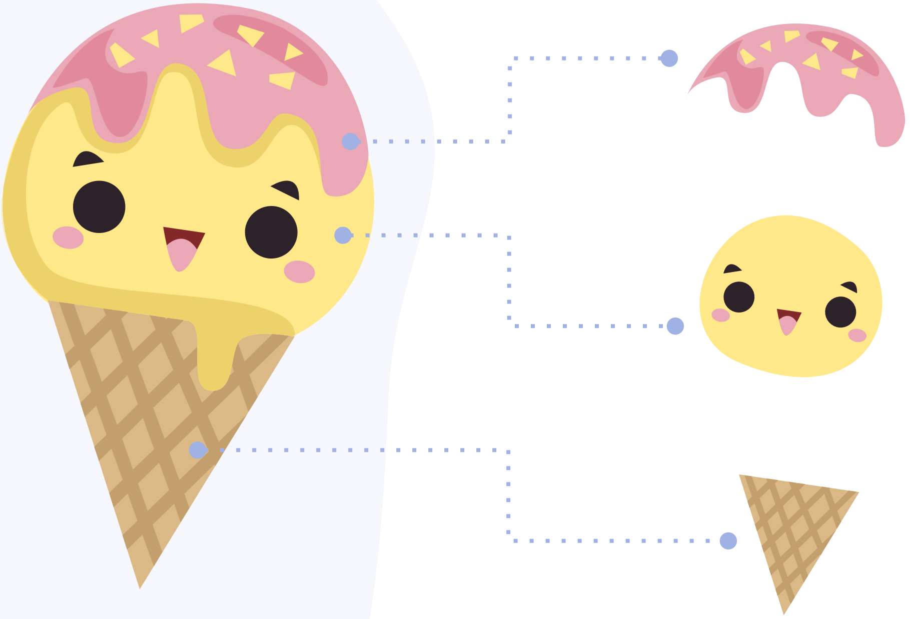
```
42["FLAG","Hell[illegible]
Websockets!!!",[illegible]32fa02048g19g941nqttoenlbqdr1ptleiuapau2bn575us0rscgrf
[illegible]8kam61t5vnv0obd0eap0llmkhtdk8ot29"]
```

History

Message	Direction	Manual	Length
2	→ To server		1
3	← To client		1
42["MESSAGE","Hello Websockets!!!","32f...	→ To server	✓	141
42["MESSAGE","Hello Websockets!!!","32f...	← To client		141
42["FLAG","Hello Websockets!!!","32fa02...	→ To server	✓	138
42["FLAG","flag{BoloRei_And_WebSocket...	← To client		48

```
42["FLAG","flag{BoloRei_And_WebSockets_4_Xmas}"]
```

Flags



WebSocket Flag

flag{BoloRei_And_WebSockets_4_Xmas}

Santa JavaScript Flag

flag{This_Flag_Is_Easier_Than_Rabanadas}

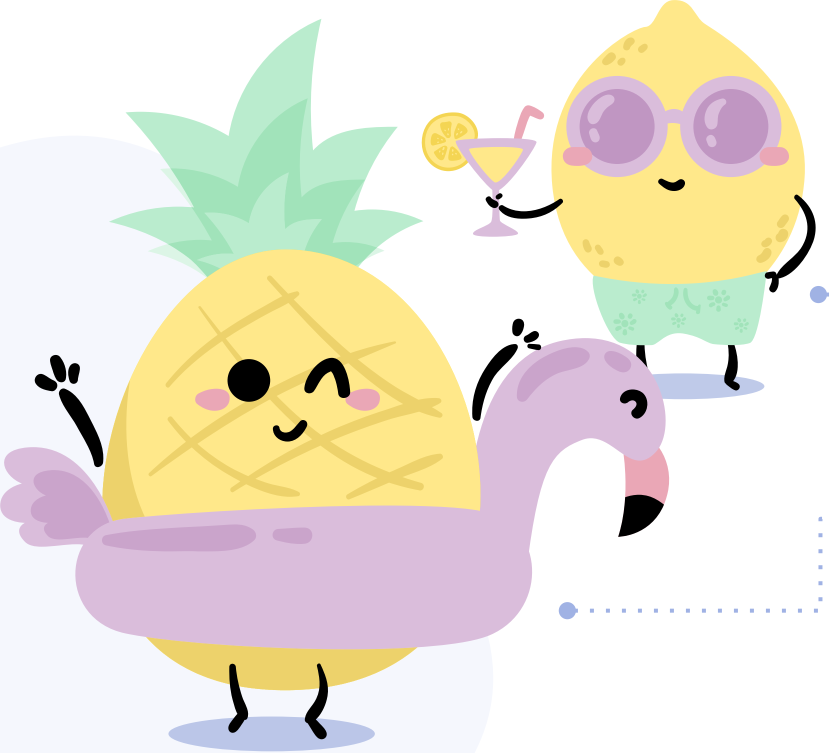
Santa Wasm Flag

flag{TheresNoXmasWithoutArrozDoce}

Solvers

- hcosta (3/3)
- sergio(3/3)
- vpinho (3/3)
- ArmySick(2/3)
- jp(2/3)
- nunohumberto(1/3)

Source



Base

https://github.com/robertDurst/ChatDemo

CTF

https://github.com/zezadas/0P0SEC-0x6C6461703A2F2F-WebChatCTF

Thank you!

Do you have any questions?

zezadas

@0xz3z4d45

https://sefod.eu

Special thanks to Inês for helping with the web development

Thank you!

Do you have any questions?

youremail@freepik.com
+34 653 090 098
Yourwebsite.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution