# Мастер-класс по поиску IDOR(BOLA)

Анна Васильева

ATI.SU

HEISENBUG

# Whoami

AppSec в ATI.SU

ati.su
биржа грузоперевозок

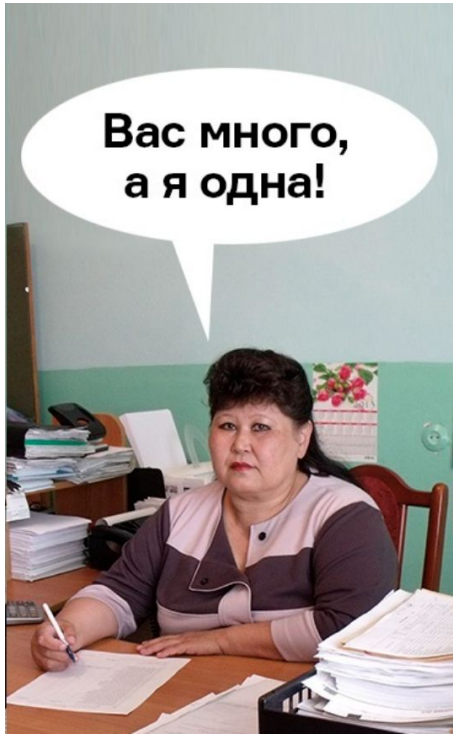
6 лет в ИТ:
Qa->Qa automation->AppSec

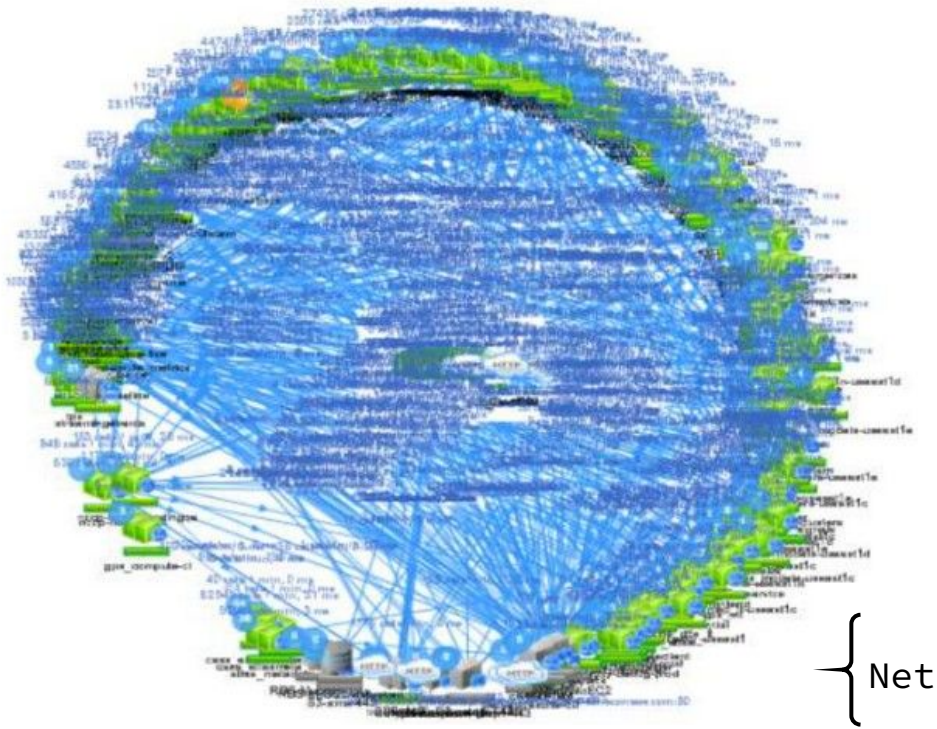
# План мастер-класса

- Обсудим цель мастер-класса
- Что такое IDOR
- Покажу с помощью Burp Suite как их искать
- Немного автоматизации - плагин Autorize
- Примеры интересных багов
- Почему возникает IDOR
- Место IDOR среди уязвимостей
- Какие есть приемы поиска Idor

# Зачем все это нужно QA?

# Соотношение AppSec к команде разработки не большое



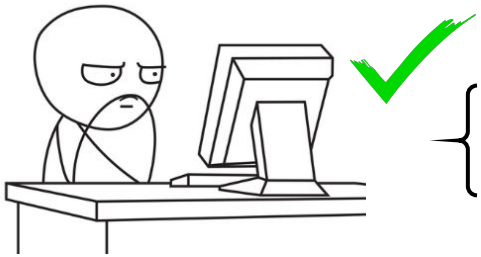



Netflix

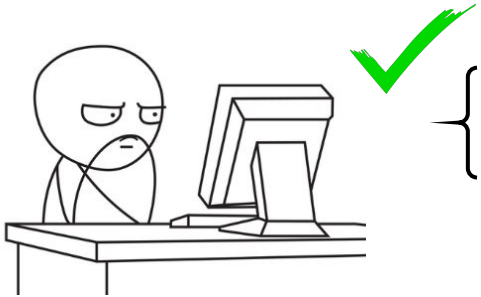
# Security champion

Security Champion — это человек внутри команды разработки, который больше всех заинтересован в безопасности продукта

# Появляется компетенция ИБ внутри команды



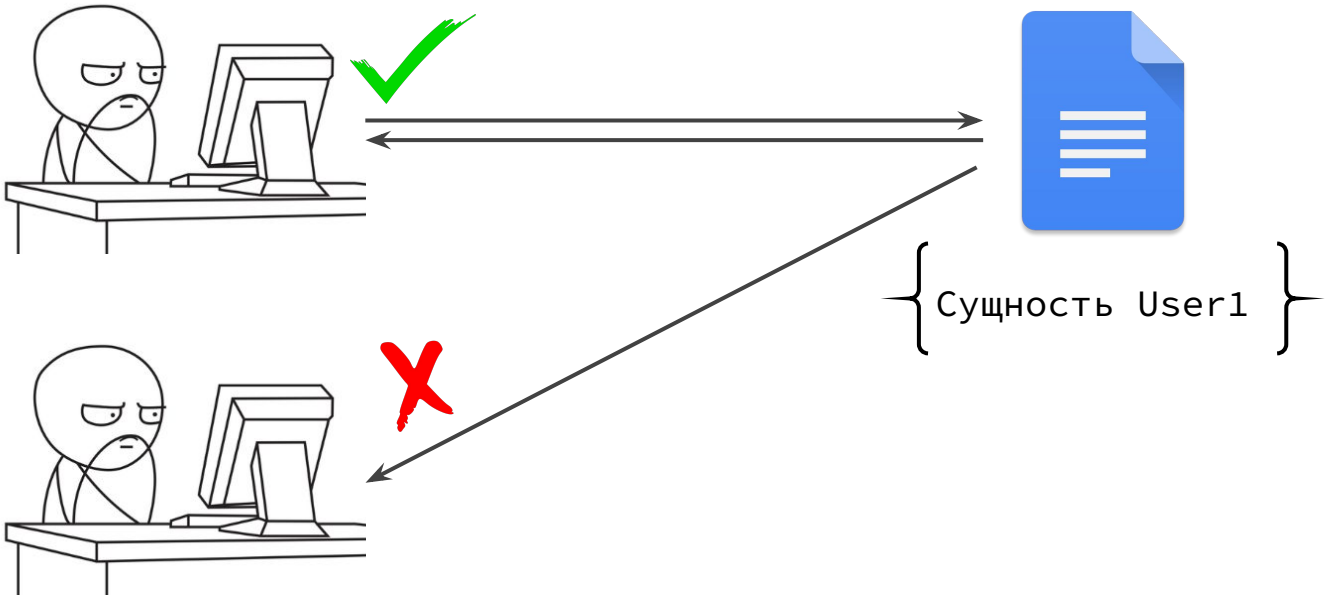
Developer смотрит на код ревью



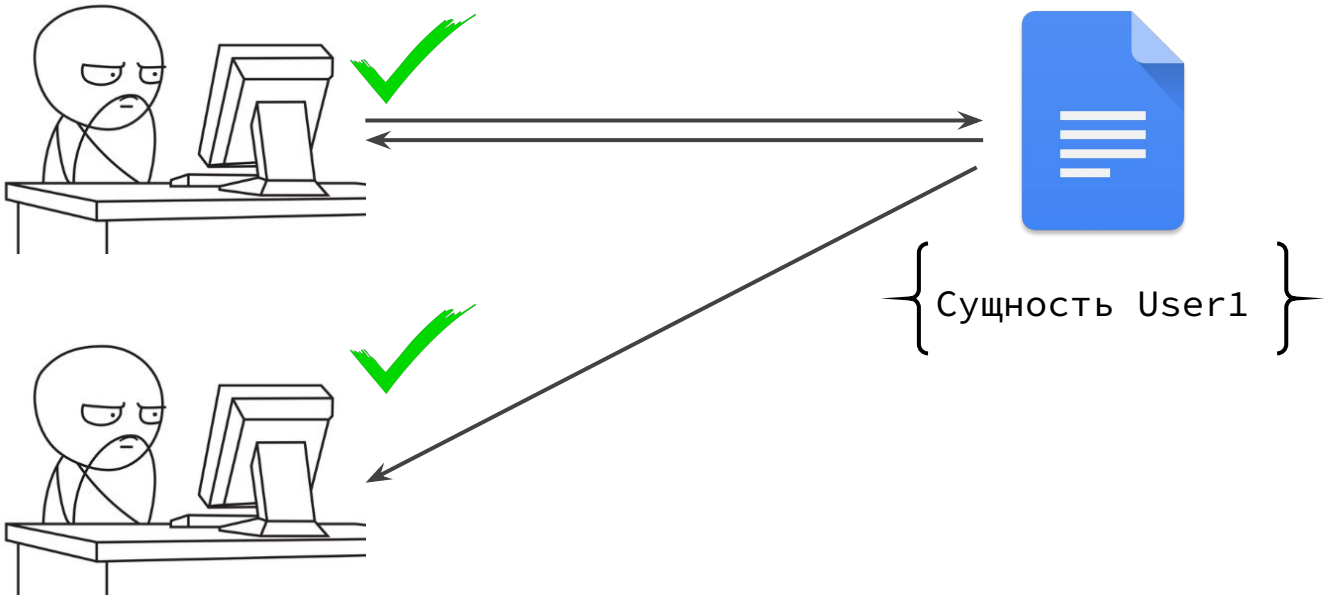
QA тестирует безопасность

# Insecure Direct Object Reference (IDOR)

# Получение документов - как должно быть



# IDOR в получении документов



# Приступим к поиску!

# Как начать - инструкция по установке Burp

- Установить и запустить Burp Suite https://portswigger.net/burp/releases#community
- Включить проксирование 127.0.0.1:8080
- Перейти на http://burpsuite/ и скачать сертификат
- Устанавливаем его в хранилище доверенных сертификатов ОС или в свое хранилище у firefox
- Скачать jython standalone https://www.jython.org/download.html
- На вкладке Extender/Options добавить его в Python Environment
- Перейти на вкладку Extender/BApp Store - выбрать слева "Autorize"- справа и нажать install

# Примеры интересных багов

# Получение всех операций чужого аккаунта

```
POST /balance

Host: test

Content-Type:
application/json-patch+json
```

```
{account_id: 1996240}
```

Получаем чужие операции всех аккаунтов

| Request | Payload | Status |
|---|---|---|
| 0 | | 200 |
| 1 | 1962397 | 200 |
| 2 | 1962398 | 200 |
| 3 | 1962399 | 200 |
| 4 | 1962400 | 200 |
| 5 | 1962401 | 200 |
| 6 | 1962402 | 200 |
| 7 | 1962403 | 200 |

# Курс валют при переводе можно задать

```
POST /currency_transfer

Host: test

Content-Type:
application/json-patch+json


{exchange_rates: 0.016,

cur_from: RUB,

cur_to: USD…}
```

```
1 рубль -> 1 доллар
```

# Можно получить бесплатно услугу, передав флаг

```
PATCH /advert

Host: test

Content-Type:
application/json-patch+json


{payment: true…}
```

**Проверить**

```
POST -> PATCH

Response -> Request

error = false
```

# Функционал доступен всем не авторизованным пользователям

```
GET /files

Host: test

Content-Type:
application/json-patch+json
```

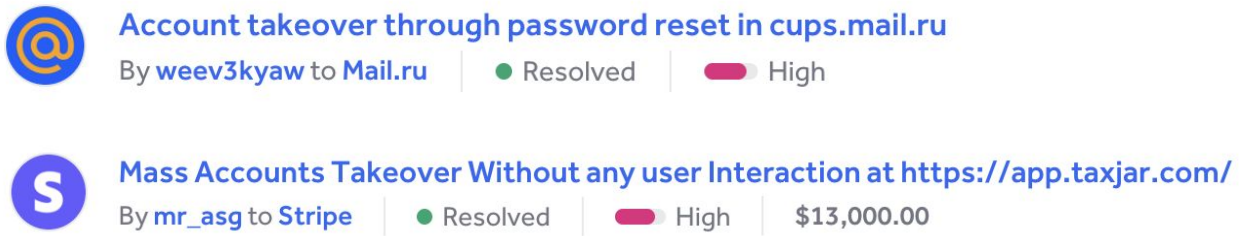
Убираете авторизационные данные и смотрите

| Authz. Status | Unauth. Status |
|---|---|
| Enforced! | Enforced! |
| Bypassed! | Bypassed! |
| Is enforced??? (ple... | Is enforced??? (ple... |
| Bypassed! | Bypassed! |
| Bypassed! | Bypassed! |

# Смена пароля у любого пользователя

```
GET /reset-password/[hash]

Host: test

Content-Type:
application/json-patch+json
```

[hash] - можно получить по публичному UserId



# Почему возникает IDOR?

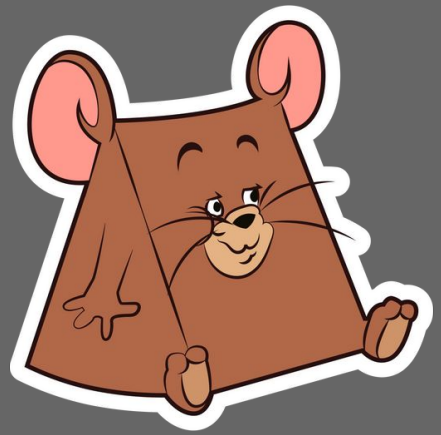
# Забыли проверить авторизацию

# Получение приватной сущности - профиля пользователя

```
user_profile =
self.profile.get(user_id)
```

```
if user.account:
    user_profile =
self.profile.get(user_id)
```

# Записываем все что пришло на входе без санитизации



# Обновление сущности

```
raw_data = await
request.json()

form = FormsModel(self._db,
raw_data)
```

```
Нужно валидировать raw_data
```

# Иногда проверки можно обойти

# Проверяется POST - обходится через GET

```
if request.post?
      req_params =
user_update_params
    end
```

Обходится через

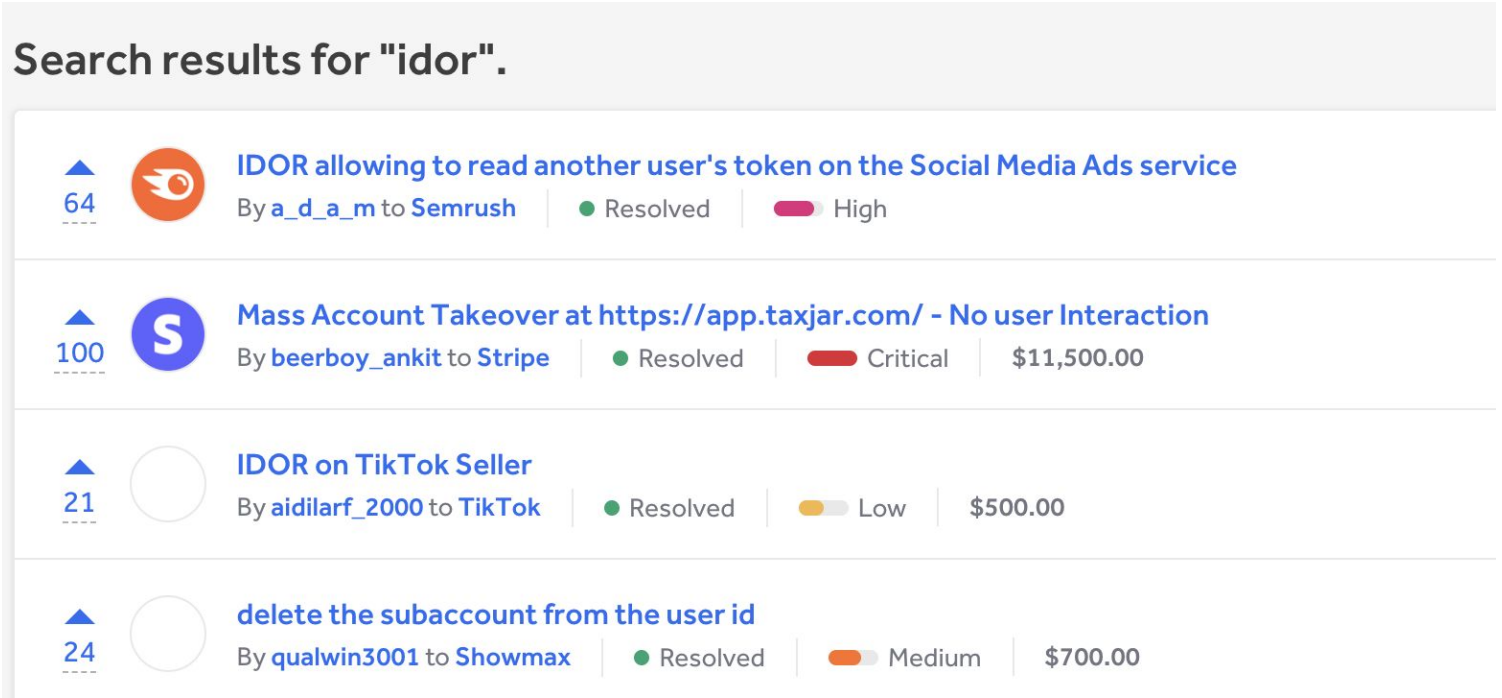
_method=GET

# Место IDOR среди уязвимостей

# IDOR в owasp api top 10: 2019

| |
|---|
| API1:2019 Broken Object Level Authorization |
| API2:2019 Broken User Authentication |
| API3:2019 Excessive Data Exposure |
| API4:2019 Lack of Resources & Rate Limiting |
| API5:2019 Broken Function Level Authorization |
| API6:2019 Mass Assignment |
| API7:2019 Security Misconfiguration |
| API8:2019 Injection |
| API9:2019 Improper Assets Management |
| API10:2019 Insufficient Logging & Monitoring |

# IDOR в репортах hackerone



# IDOR vs BOLA?

# Insecure Direct Object Reference
# =
# Broken Object Level Authorization

# Чек-лист с приемами по поиску IDOR

# Итоги

## IDOR легко тестируется

- Подставляй чужие id
- Включай Autorize и ходи по сайту 😏

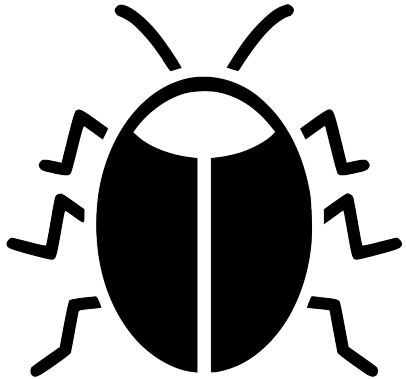
## IDOR легко автоматизируется

```
@allure.id("11399")

@allure.title("Нельзя удалить чужой файл")

def test_cant_delete_another_file(self, api,
deal, other_file_jpg):
```

# Всем спасибо за внимание!

# Вопросы?

{ QA освоивший пентест }



{ https://t.me/iSavAnna }