



ПРОВЕДЕНИЕ ИССЛЕДОВАНИЙ В СЕТИ ИНТЕРНЕТ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

МОДУЛЬ 2

ЛЕКЦИЯ 6

ОСНОВЫ РАБОТЫ С РЕГУЛЯРНЫМИ ВЫРАЖЕНИЯМИ И ПРОИЗВОДСТВО НТТР- ЗАПРОСОВ. БИБЛИОТЕКА РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ RE

*Лектор:
к.ф.-м.н., доцент кафедры
программного обеспечения вычислительной техники и
автоматизированных систем*

Зуев С.В.


ПОНЯТИЕ РЕГУЛЯРНОГО ВЫРАЖЕНИЯ

PATTER

Уязвимос
Т

Метасимвол

\backslash \$ * + ? { } [] \ | ()

Б.{4}род 

Белгород
Б.огород
Б. Народ
Безпород

В мае издание [Financial Times](#) сообщило об использовании хакерами [уязвимости](#) нулевого дня в [WhatsApp](#), позволявшей подслушивать пользователей, читать их зашифрованную переписку, включать микрофон и камеру и устанавливать шпионское ПО для дальнейшей, более глубокой слежки. Чтобы воспользоваться [уязвимостью](#), злоумышленник мог просто позвонить жертве по [WhatsApp](#). Особая конфигурация вызова провоцировала переполнение буфера клиента [WhatsApp](#) и открывала атакующему возможность перехватить контроль над приложением и выполнить из него произвольный код. По-видимому, злоумышленники прибегали к этому методу не только для прослушивания звонков и перехвата переписки пользователей, но и для установки своих приложений на устройства за счет ранее [неизвестных уязвимостей](#) в операционных системах. Разработчики [WhatsApp](#) быстро выпустили исправление, и на этом, казалось бы, все закончилось. Однако в октябре компания [подала иск](#) против израильской фирмы [NSO Group](#), обвиняя последнюю в создании эксплоита. [WhatsApp](#) заявила, что технология, продаваемая NSO, была использована для атаки на мобильные телефоны более 1400 пользователей в 20 странах, в числе которых были правозащитники, журналисты и другие общественные деятели. NSO все обвинения отрицает.

В июле мы опубликовали закрытый отчет о последних версиях [FinSpy](#) для [Android](#) и [iOS](#), разработанных в середине 2018 года. Создатели [FinSpy](#) продают свое ПО правительственным и правоохранительным органам по всему миру, помогая им собирать конфиденциальную информацию пользователей на различных платформах. При этом мобильные [импланты](#) для [iOS](#) и [Android](#) реализованы схожим образом. Они позволяют собирать личные сведения, такие как контакты, сообщения, электронные письма, календари, [GPS](#)-координаты, фотографии, файлы в памяти устройства, записи телефонных разговоров и данные из наиболее популярных мессенджеров. С помощью [импланта](#) для [Android](#) можно также получить [root](#)-привилегии на [нерунованном](#) устройстве, используя известные [уязвимости](#). [Имплант](#) для [iOS](#), по всей видимости, не предоставляет возможностей для заражения, но помогает удалять следы использования общедоступных инструментов для [джейлбрейка](#). А значит, программе необходим физический доступ к устройству жертвы, если оно еще не взломано. Последняя версия ПО включает целый ряд ранее неизвестных функций. В ходе нашего недавнего исследования мы обнаружили актуальные версии [имплантов](#) более чем в 20 странах, хотя размер клиентской базы позволяет предположить, что реальное количество жертв гораздо больше.

МЕТОДЫ И ФУНКЦИИ БИБЛИОТЕКИ re

ФУНКЦИЯ match, МЕТОДЫ group(), groups()

```
re.match(pattern, string,  
flags=0)  
import re  
title = "Белгород 308000, ул. Костюкова, 46"  
example = re.match( '(.*)(\d{6}), (.*)', title,  
re.M|re.I)  
if example:  
    print("example.group(): ",  
example.group())  
    print("example.group(1): ",  
example.group(1))  
    print("example.group(2): ",  
example.group(2))  
    print("example.groups(): ",  
example.groups())  
else:  
    print("Нет совпадений!")
```

```
Белгород 308000, ул. Костюкова, 46  
Белгород  
308000  
( 'Белгород', '308000', 'ул. Костюкова, 46')
```

МЕТОДЫ И ФУНКЦИИ БИБЛИОТЕКИ re

ФУНКЦИИ search И findall

```
re.search(pattern, string,  
flags=0)  
title = ""Белгород 308000 ул. Костюкова, 46  
Москва 127000, ул. Академика Королева,  
12""
```

Ищем пробел перед
строкой

```
example = re.search( '^(.*) (\d{6}), (.*)', title,  
re.M|re.I)
```

```
if example:
```

```
    print("example.group():",  
example.group())  
    print("example.group(1):",  
example.group(1))  
    print("example.group(2):",  
example.group(2))
```

```
    print("example.group(3):",  
example.group(3))
```

```
    print("example.groups():",
```

```
Москва 127000, ул. Академика Королева, 12  
Москва  
127000
```

```
ул. Академика Королева, 12
```

```
('Москва', '127000', 'ул. Академика Королева, 12')
```

Функция findall объединяет match и search: ищет pattern по всей

строке

МЕТОДЫ И ФУНКЦИИ БИБЛИОТЕКИ re

МЕТОД sub

```
re.sub(pattern, repl,  
string)
```

Заменяет в строке string вхождение pattern на repl

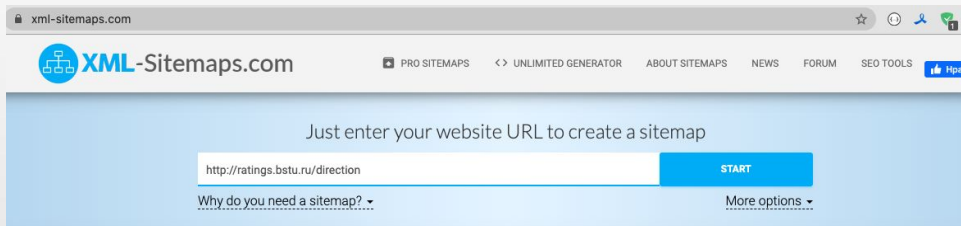
```
title = ""Белгород 308000 ул. Костюкова,  
46 Москва 127000, ул. Академика  
Королева, 12""  
ex2 = re.sub('\d{6}', '111111', title)  
print(ex2)
```

```
Белгород 111111 ул. Костюкова, 46  
Москва 111111, ул. Академика Королева,  
12
```

HTTP-ЗАПРОСЫ С РЕГУЛЯРНЫМИ ВЫРАЖЕНИЯМИ

Формируем XML-карту сайта по адресу `http://ratings.bstu.ru/direction` с помощью инструмента `xml-sitemaps.com` – кладем ее в файл

```
from bs4 import BeautifulSoup as BeSo
import requests
import re
content = []
with open("../rbstu-sitemap.xml", "r") as file:
    content = file.readlines()
    content = "".join(content)
    bs_content = BeSo(content, "lxml")
add_list = []
pbs = bs_content.findAll('loc')
for p in pbs:
    p1 = p.renderContents().decode('UTF-8')
    pr = re.findall('http://ratings.bstu.ru/direction/000000003/000000001/00000000\d{1}/09.*', p1) if
pr:
    add_list.append(pr)
for j,_ in enumerate(add_list):
    response = requests.get(add_list[j][0])
    soup = BeSo(response.text)
    P = soup.find(text='По общему конкурсу').findNext('tbody').findAll('tr') counter = 0
    for p in P:
        Yes = p.findAll('td')[4].renderContents().decode('UTF-8') if Yes == 'Есть':
            counter+=1
```



ОТВЕТ НА ЗАПРОС

Имеется	40	согласий на направление	09.03.01
Имеется	45	согласий на направление	09.03.02
Имеется	21	согласий на направление	09.03.03
Имеется	30	согласий на направление	09.03.04
Имеется	10	согласий на направление	09.04.01
Имеется	11	согласий на направление	09.04.01
Имеется	13	согласий на направление	09.04.02
Имеется	12	согласий на направление	09.04.04

Результат пригоден для машинного применения, но еще не визуализирован