SSRF DoS Relaying and a bit more...

Alexander Bolshev
@dark_k3y

15th DefCon Russia meeting

March 27, 2013

; cat /dev/user

Digital Security

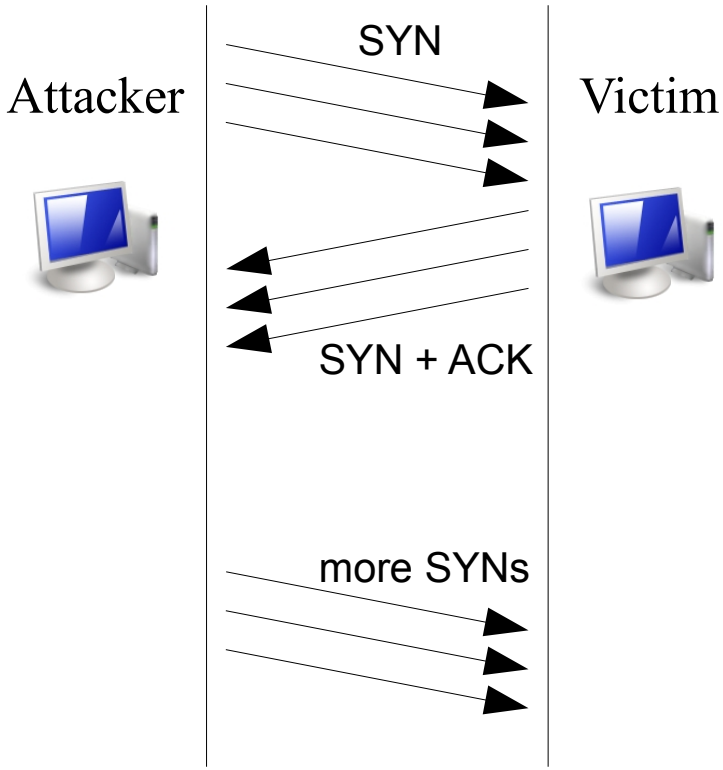
Alexander Bolshev aka @dark_k3y

- IS auditor @ Digital Security
- Ph.D.
- just another man with somecolorhat

Digital Security
Research Group

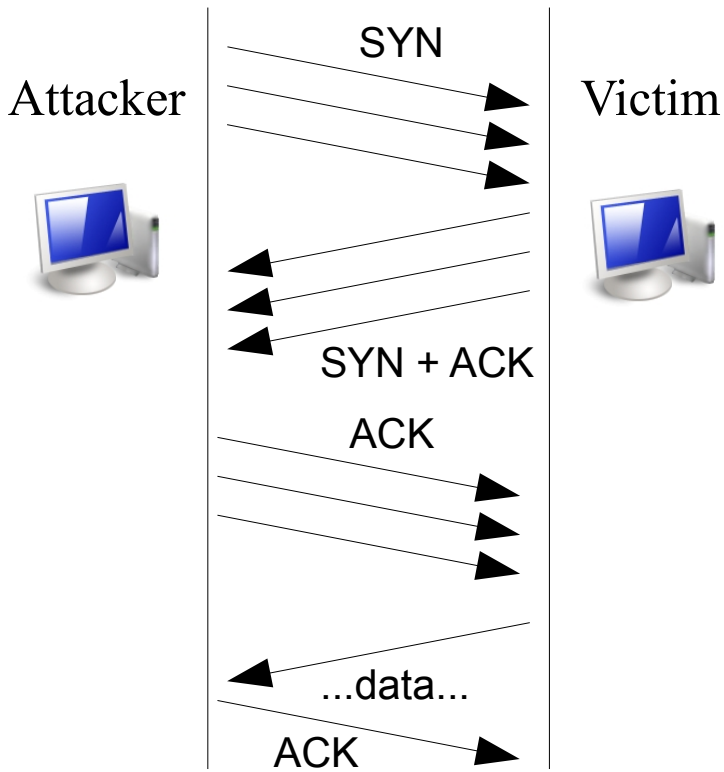
Back to 199x: SYN- and connect- DoS attacks

Half-opened DoS



TCP only resources exhaustion, easy to defend

Full-opened DoS



TCP **and service** resource exhaustion, much harder to defend

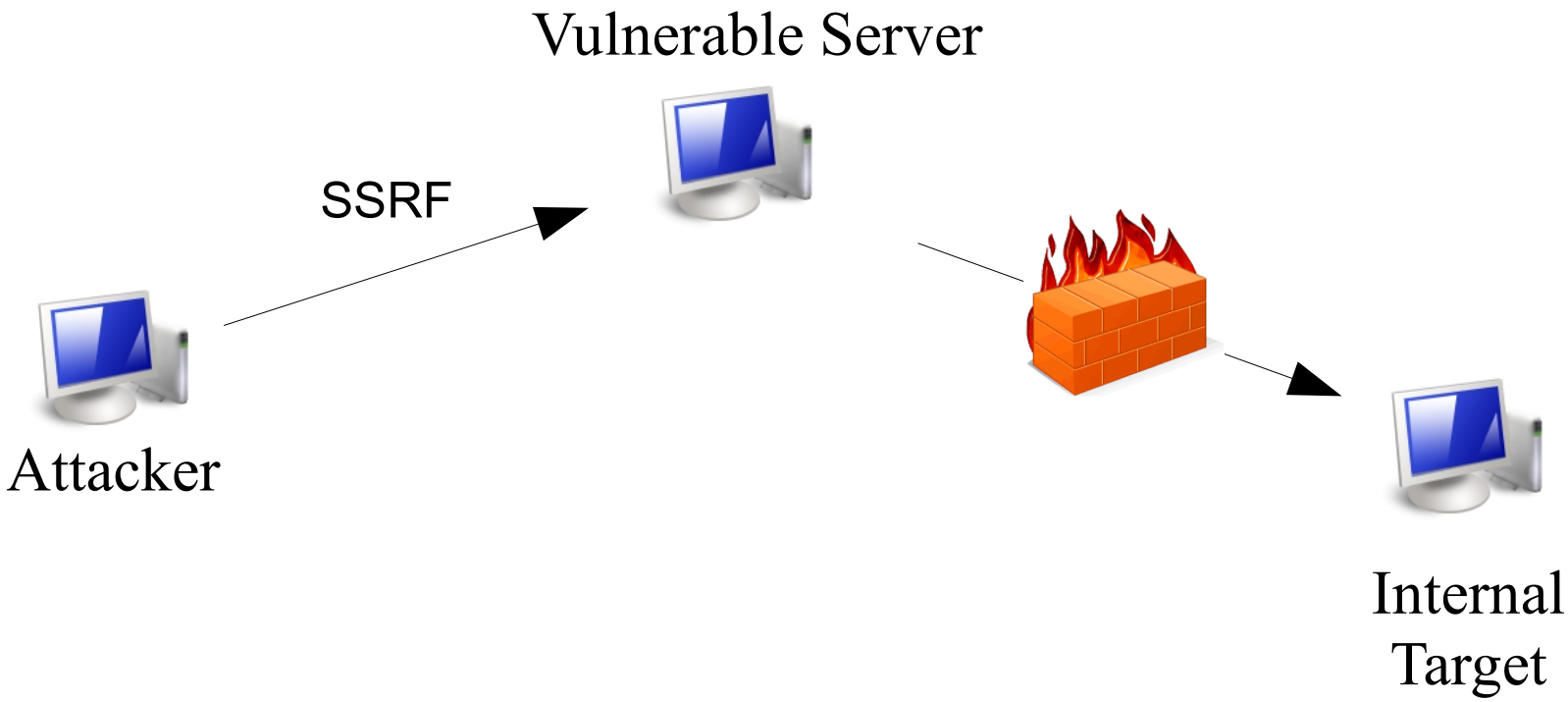
Full-opened connection DoS

- Full-opened connection DoS is much more effective against target.

...but!

- Full-opened connection DoS with data exchange requires much more resources on attacker host.
- Full-opened connection DoS cannot be spoofed (in general case).

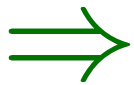
All you know what is SSRF(hopefully).



Classic SSRF attack scheme

Idea!

We can relay connections with SSRF



We can relay full-opened DoS with it!

But you can't just simply relay it... [1] [2]

You can't just simply DoS with pure SSRF.

[1] [||||||], but I couldn't resist to place this pic

[2] *sometiMe$* you can...

You can't just simply relay it... Why?

Full-opened DoS attack with pure SSRF will be ineffective cos of:

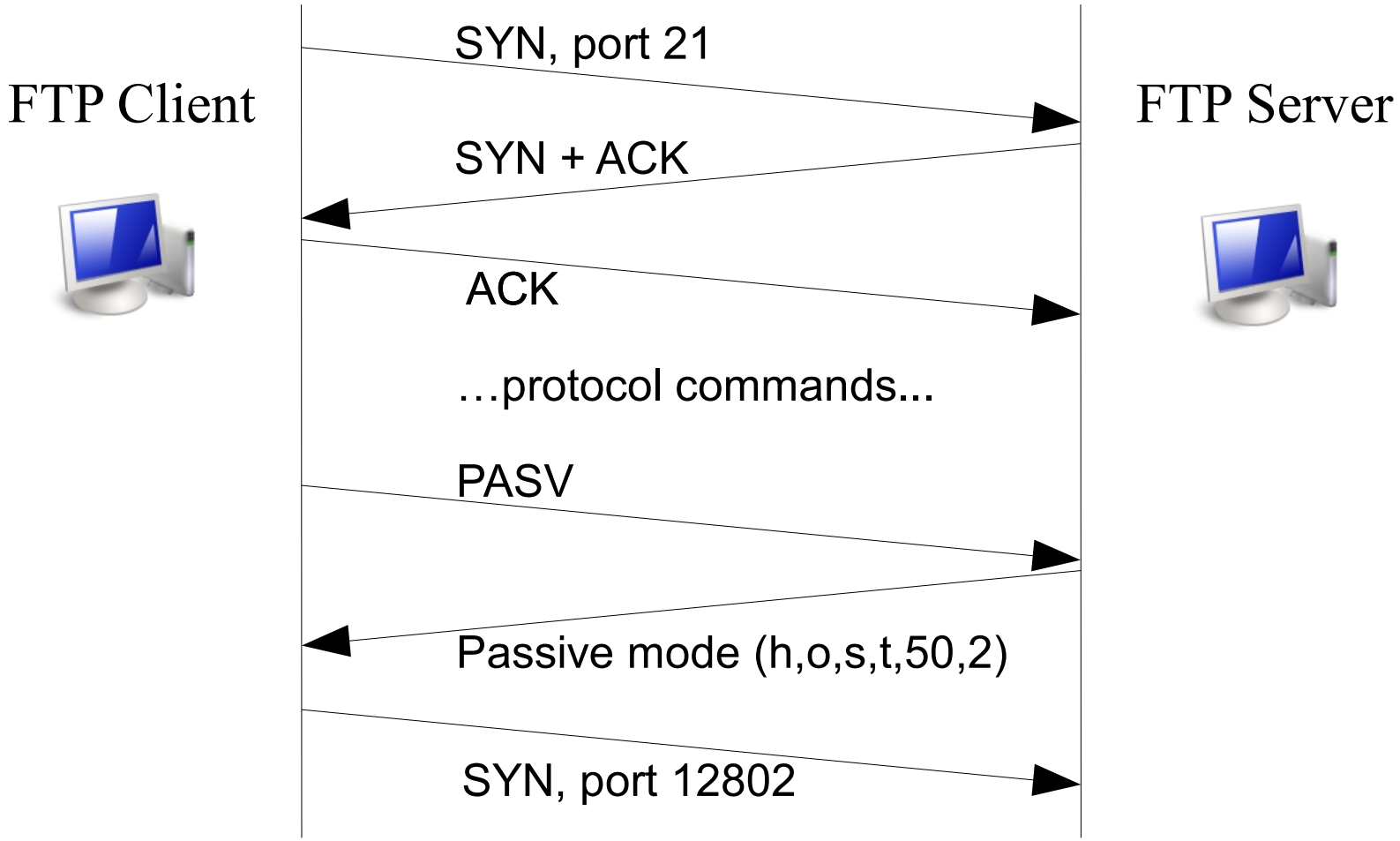
- You should hold opened connection with the relay, while relay holds connection with the victim.
- Most protocols drops connection when you're using invalid format (HTTP && others).
- You're dependant on your host capacity, not on relay host capacity.

But we have FTP!

Almost all "SSRF-enabled" technologies support FTP URI scheme.
The FTP for relaying is interesting cos of:

- FTP has two connections between client and server: **control** and **data**.
- While control connection may be closed, data connection will exists till the end of "transaction" or timeout.
- FTP passive mode allow to exact specific remote port(**!**) and host (**!!**) for data connection.

FTP passive connection scheme



Difference between PASV and EPSV

PASV:

- old version of FTP, allows to establish control connection to any host/port.

EPSV:

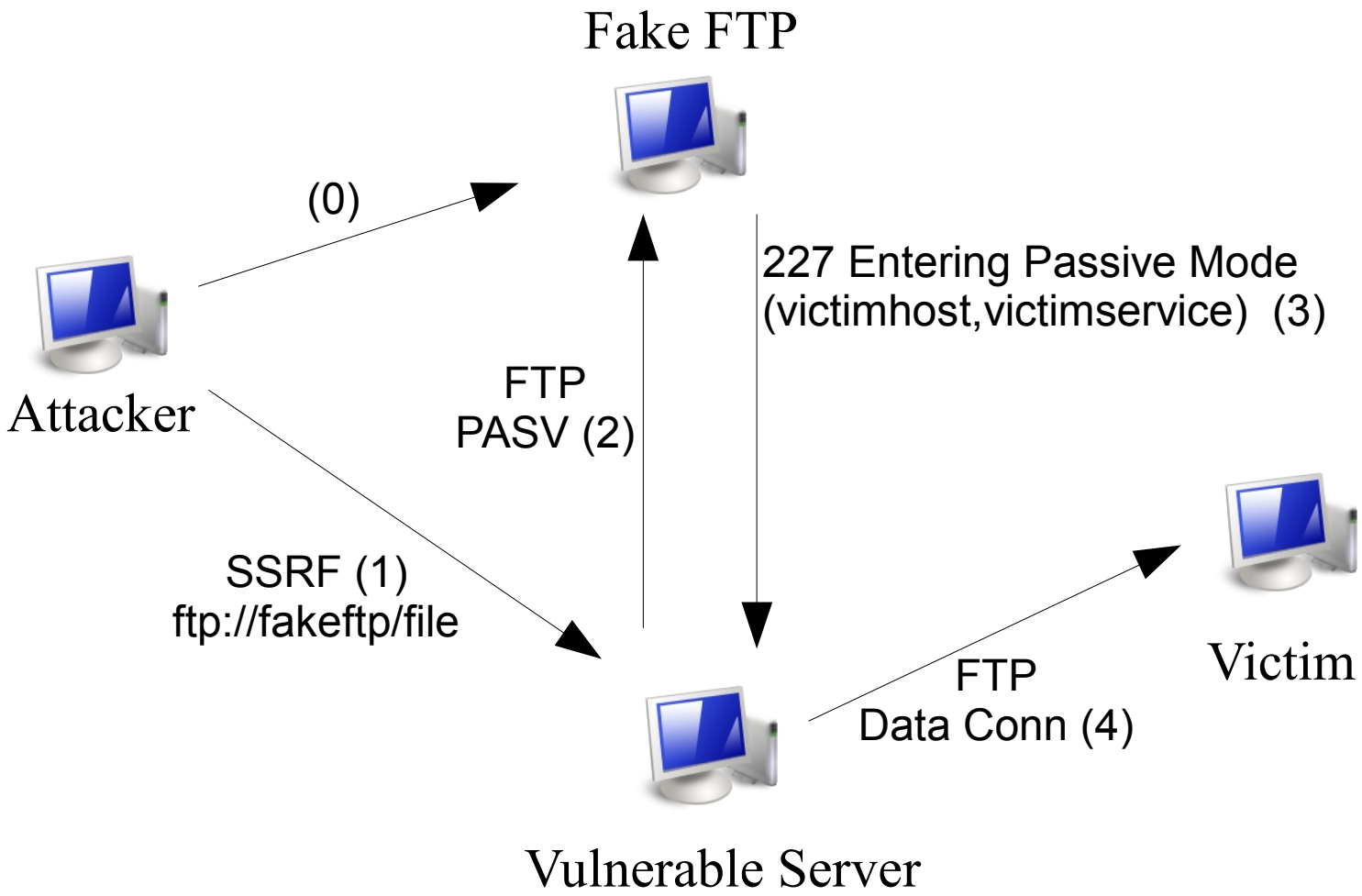
- modern versions of FTP, allows to establish control connection only to specific FTP server port.

So we can't use PASV anymore?

We just can say that we didn't support it!

```
220 i58 FTP server ready.
USER anonymous
331 Guest login ok, send your email address as password.
PASS Java1.6.0_01@
230 Guest login ok, access restrictions apply.
TYPE I
200 Type set to I.
EPSV ALL
500 Command not implemented, superfluous at this site.
PASV
227 Entering Passive Mode (vic,tim,server,ip,0,80).
RETR doc
150 Opening BINARY mode data connection for 'doc' (99999 bytes).
```

Attack scheme



Attack inside wireshark

192.168.200.138	192.168.200.128	FTP	76	Request: EPSV ALL
192.168.200.128	192.168.200.138	FTP	121	Response: 500 Command not implemented, superfluous
192.168.200.138	192.168.200.128	FTP	72	Request: PASV
192.168.200.128	192.168.200.138	FTP	110	Response: 227 Entering Passive Mode (46,4, ,0,80
192.168.200.138	46.4.	TCP	74	36659 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 S
192.168.200.138	192.168.200.128	TCP	66	46376 > ftp [ACK] Seq=61 Ack=251 Win=14720 Len=0 TS
46.4.	192.168.200.138	TCP	60	http > 36659 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
192.168.200.138	46.4.	TCP	54	36659 > http [ACK] Seq=1 Ack=1 Win=14600 Len=0
192.168.200.138	192.168.200.128	FTP	77	Request: RETR file
192.168.200.128	192.168.200.138	FTP	134	Response: 150 Opening BINARY mode data connection f
192.168.200.128	192.168.200.138	TCP	66	ftp > 46376 [FIN, ACK] Seq=319 Ack=72 Win=64169 Len

Attacking HTTP server on remote host (46.4.x.x) with 192.168.200.138 relay using Fake FTP on 192.168.200.128.

Knowing technology "features" makes you stronger

Technology	Ability to relay DoS with FTP
PHP	Yes[1]
cURL	Yes
LWP	Yes[1]
Java $\leq$ 1.6.x	Yes[1]
Java 1.7.x	Partially[1] [2]
ASP.Net	No [3]
Python	Yes

[1] DoS with FTP control connection available

[2] only supports data connections to localhost and FTP server address.

[3] but as always has a "killer feature"...

Java trying to defend: Fail!

In Java 1.7.x devlopers tried to mitigate this "feature" by disabling data connections to any other hosts except FTP server.

But they forgot to disable data connections to localhost!

ASP.Net: Much More FAIL.

ASP.Net don't support PASV command. Only EPSV.

..but..!

... when an XXE injection is executed, control FTP connection to the remote host is established **in any case**. This connection is **sustained** after the termination of the SSRF attack connection.

Possible mitigations

On administrators side:

- Disable ALL non-established outgoing TCP packets from host (on ALL ports, even on TCP). Hard to do, more problems, much pain. ☹

On developers side:

- Don't make mistakes that lead to SSRF. (cap is laughing here)

On vendors side:

- Disable PASV command **at all** (because there are no more FTP servers that don't support EPSV). But to it in *another way* than Oracle and Microsoft.

Back to 199x again: other funny stuff (not mine)

There're more techniques that exploits the client side of SSRF:

- libcurl SASL buffer overflow vulnerability (by Volema, see CVE-2013-0249) [1]
- port scanning like FTP BOUNCE (hello, Fydor!) but with SSRF (several whitepapers on the internet) and more (google for Vladimir Vorontsov "SSRF Bible cheatsheet")

[1] http://curl.haxx.se/docs/adv_20130206.html

More fun – some bombs and ldap freeze

Another cool stuff to do:

- gzip bombs in PHP and LWP (PHP didn't support gzip-compression in HTTP, **but...** FAIL! compress.zlib://http:// and zlib://http:// are your friends).

- Have you ever tried to search the base DN with a filter of userid=* (or similar) with a SORT on userid on LDAP server with 10k users? $\Rightarrow$ DoS on client side and freeze on server side: **double strike!**

; lookman

SSRF DoS Relaying article:

- http://habrahabr.ru/company/dsec/blog/171549/ [RUS]

; thanksgiving

Thanks to:

- Alexander Polyakov aka @sh2kerr for the idea
- Vladimir Vorontsov aka @d0znpp for the SSRF Bible cheatsheet
- Fedor Savelyev aka alouette for some good thoughts about PHP
- defcon 7812 :)

Q & A

Questions?