Forschungszentrum Jülich
in der Helmholtz-Gemeinschaft

OpenGridForum
OPEN FORUM | OPEN STANDARDS

Firewalls and Grids
Experimental solution ideas

E.Gruenter@fz-juelich.de
M.Meier@fz-juelich.de
R.Niederberger@fz-juelich.de

GGF 18 - FI-RG - 13.09.2006

Overview

- **GFCP –**
  The Grid Firewall Communication Protocol

- **FUHP –**

  Firewall UDP Hole Punching

- **FSIP –**

  The Firewall Session Initiation Protocol

Grid requirements

- A Grid is a union of geographically distributed, independent organizations
- Dynamic use of resources, often in parallel

The initial problem:

- Internal hosts are protected by local firewalls
- Often only outgoing connections are allowed
- Having a client and server model implies one of both has to have an incoming connection
- So none can start communication

Solution requirements

- Integration in existing security concept
- Usable in open source and commercial environments
- Communication between partners only for minimum necessary duration

GFCP –
The Grid Firewall Communication Protocol
-
Bachelor Thesis of E.Gruenter

The ftp problem with Grids

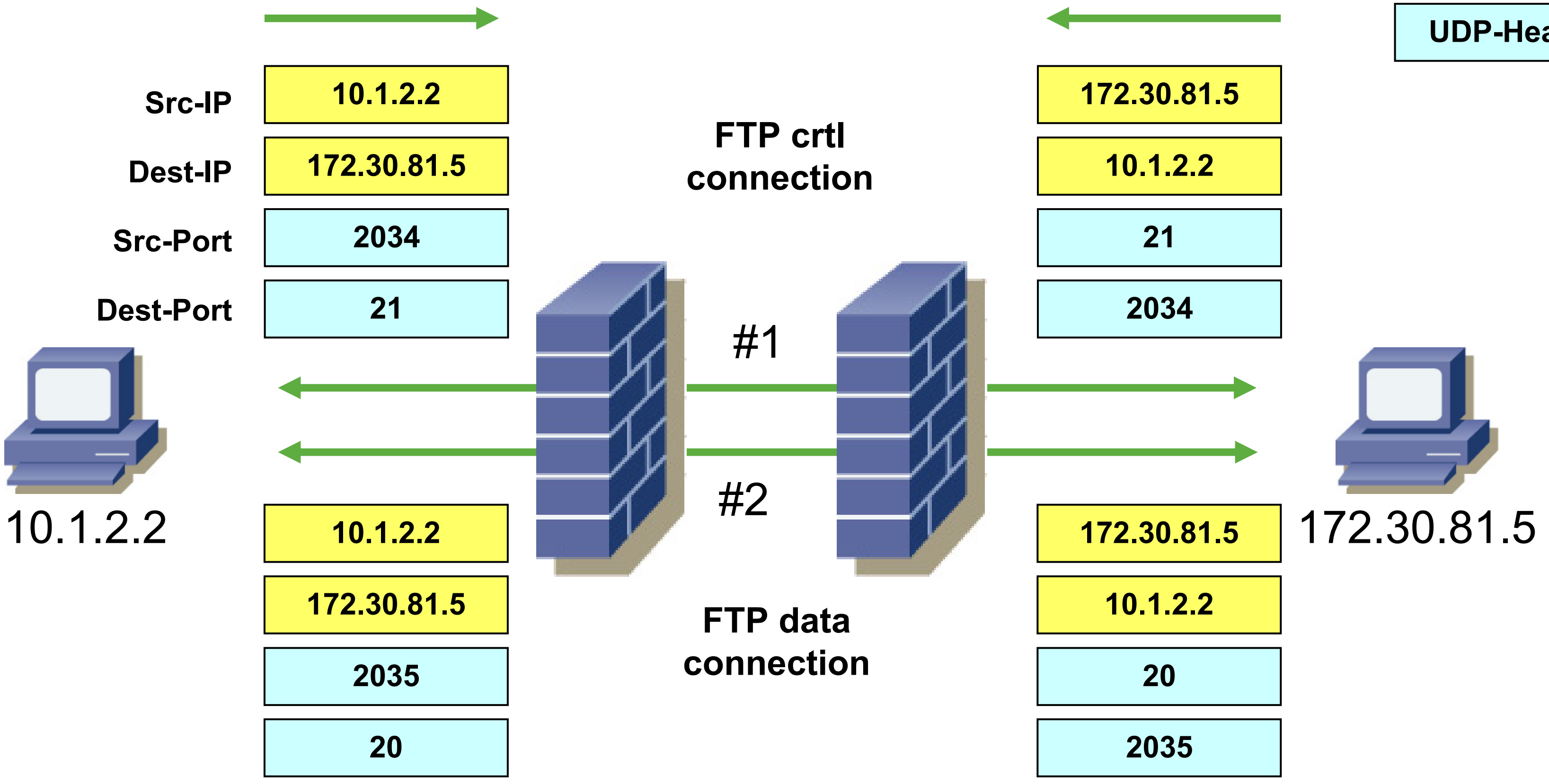
ftp provides a means to open dynamically data conns between two distributed nodes, but …

- ftp crtl conn not useable standalone for dynamic opening of ports
- ftp UID and PW are sent in clear text

The ftp problem with Grids

The initial problem: Firewalls and FTP

🟨 IP-Header
🟦 UDP-Header



FTP crtl connection

	→	←
Src-IP	10.1.2.2	172.30.81.5
Dest-IP	172.30.81.5	10.1.2.2
Src-Port	2034	21
Dest-Port	21	2034

FTP data connection

	←	→
Src-IP	10.1.2.2	172.30.81.5
Dest-IP	172.30.81.5	10.1.2.2
Src-Port	2035	20
Dest-Port	20	2035

The solution

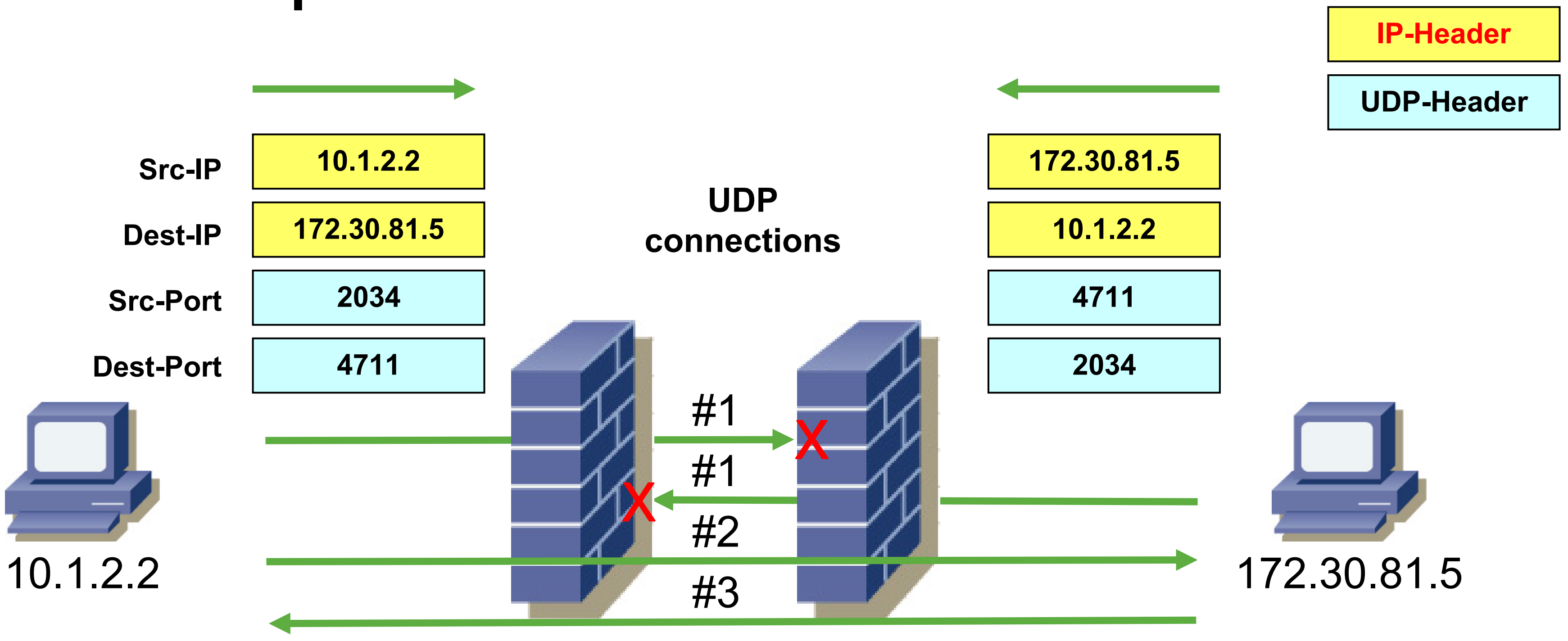
- Use ftp control streams and "FW ftp application inspection" for dynamic opening of ports
- enhance security mechanisms of ftp
- isolate/separate "ftp" crtl and data conns
- connect to GFCP server using "ftp like crtl" conn with UID: Grid and PWD: gridacc
- open "put (client → server)" data connection containing encrypted authentication information
- if authorization denied → server closes crtl conn,

  otherwise → proceed with real data conn

FUHP - Firewall UDP Hole Punching

Filtering of traffic

The initial problem: Firewalls and UDP



Src-IP	10.1.2.2	172.30.81.5
Dest-IP	172.30.81.5	10.1.2.2
Src-Port	2034	4711
Dest-Port	4711	2034

IP-Header

UDP-Header

Neither Client nor Server can reach the other one (#1)

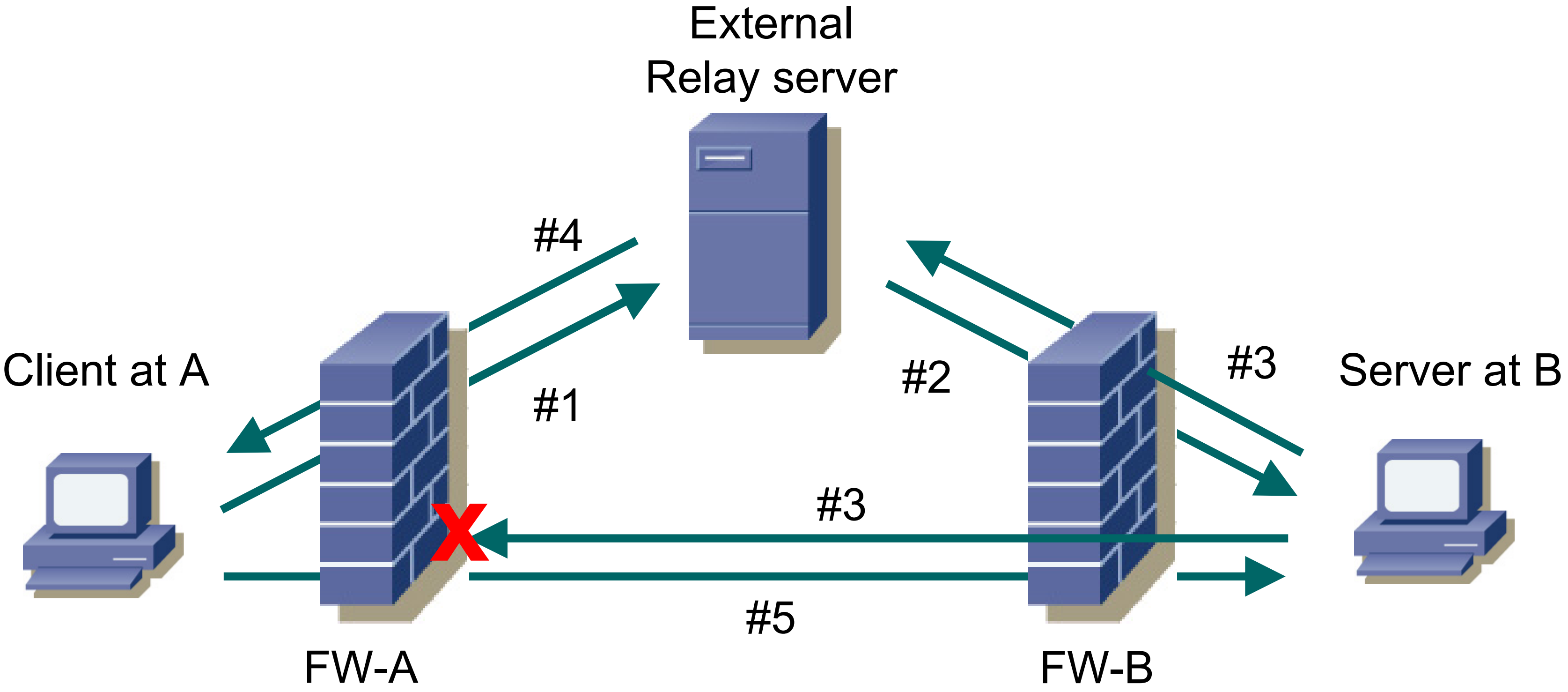
After one of both has initiated and the other knows about this, he can answer (#1, #2, #3)

first solution

- put server outside both firewalls
- harden OS system and allow only specific communication ports
- this server has crtl connections to client and server
- after having checked authenticity and authorization,
  outside server tells inside server about connection request
  from client (including client-ip and client-port info)
- inside server initiates connection to client using client-ip and -port info
  -> firewall at server side allows outgoing connection
  -> firewall at client side rejects connection
- additionally, client now connects to server, but gets through firewall
  at server side (server already opened this hole), because firewall
  at server side assumes packets from client to be answers to
  connection initiated by server

Dynamic configuration of Firewalls

The UDP hole punching concept



Simple solution, works quite well, but …

external relay server needed

- bastion host
- who administrates this server (OS and security)?
- for every service/every installation one server?

- outgoing connections have to be allowed
- works only with UDP (TCP sequence number problem)
- where is checked what and who is allowed (external or internal)?
- relay server has to handle double traffic rate per connection
- relay server has to handle multiple connections in parallel
- tables of known services have to be managed at outside server
- generalization ? (ip addr. of external servers have to be well known)

Simple improvements, now it‘s working better …

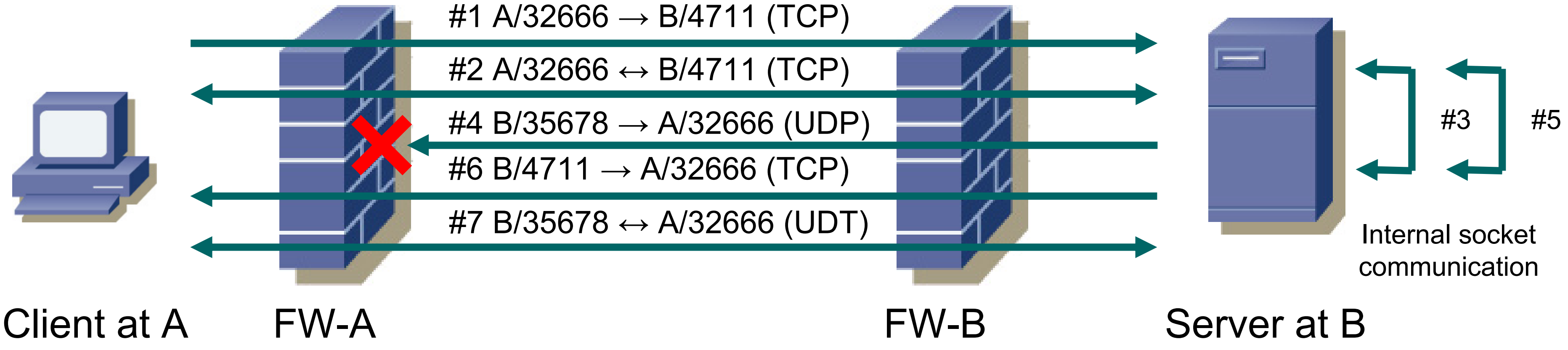
Solutions

- Combine external server and internal service at one internal host
- open well known port, e.g. TCP 4711 to access relay server
- encrypted communication between client and relay server
- internal communication between relay server and service
- check service dependent internally: authentication & authorization
- outgoing connections have to be allowed (further on required)
- works only with UDP → UDT (UDP-based Data Transfer Protocol)

Dynamic configuration of Firewalls

The UDP hole punching concept

in Grid environments


#1 A/32666 → B/4711 (TCP)
#2 A/32666 ↔ B/4711 (TCP)
#4 B/35678 → A/32666 (UDP)
#6 B/4711 → A/32666 (TCP)
#7 B/35678 ↔ A/32666 (UDT)
#3
#5
Internal socket communication
Client at A
FW-A
FW-B
Server at B

FSIP
–
The Firewall Session Initiation Protocol

Problems with GFCP

- advanced protocol handling of firewall needed (application inspection)
- allow FTP like protocol
  (Crtl conn ≠ port 21) to differentiate
  between FTP and GFCP
- Does FW allow one crtl conn with multiple data conns?
- currently only software based solution

Problems with FUHP

- allow „well known port“ FUHP
- allow UDP outgoing connections
- one crtl conn for every data conn, but could be modified
- deny messages at client FW (IDS problem?)
- Currently only software based solution

Future solution: FSIP

- Should be well known and documented

- ➢ Well defined packet format
- ➢ Fixed packet structure (hardware codeable)
- ➢ Well defined connection states (init, check, allow, deny)
- ➢ Standardize → GWD, RFC, …

Future solution: FSIP (2)

- Early stage software solution (appl. inspection)
- Should be hardware implementable in future

➢ FW life cycles prevent early deployment

→ Easy integration into available FWs as application inspection after standardization

➢ Long term: hardware (chip) solution within FW (optional for high speed)

Future solution: FSIP (3)

- Overhead should be minimized:
  → crtl conn with many data conns possible (e.g. port range)
- *Allow A to initiate data comm between B ↔ C: problematic issue*

➢ Check once, allow multiple

→ single sign on scheme

➢ No problem with normal applications, but single sign on needed for grid apps (gridFTP, metacomputing)

Future solution: FSIP (4)

- **Must be secure**
  - encrypted UID and PWD, certificates, …
  - clear text information (FW readable) and encrypted info for security
  - If "clear text information " and "encrypted info" differ

    → server closes connection → deny
  - Global principle: no crtl conn → no data conn

    → FW may have to terminate active sessions
  - Timeout for crtl conns required

    → crtl conn has to be hold active

Summary

We have it all,

so let‘s start

Questions and discussion

Questions
and
discussion