



Where .com meets .org

Berlin
27 December 2004

Workshop: Reverse Engineering the SAP R/3 Client Protocol

...for every business

Nils Magnus
Jochen Kellner

21C3 Chaos Communication Congress
Berlin, Germany
December 27 - 29, 2004

Agenda

**Overview of the SAP R/3 architecture
(from a networker's point of view)**

Problem of undocumented client protocol

Current findings

Workshop: reverse protocol details

Agenda

Why SAP R/3 should bother all of us

**Overview of the SAP architecture
(from a networker's point of view)**

Problem of undocumented client protocol

Current findings

Workshop: reverse protocol details

The SAP R/3 universe

First of all: SAP is huge and confusing

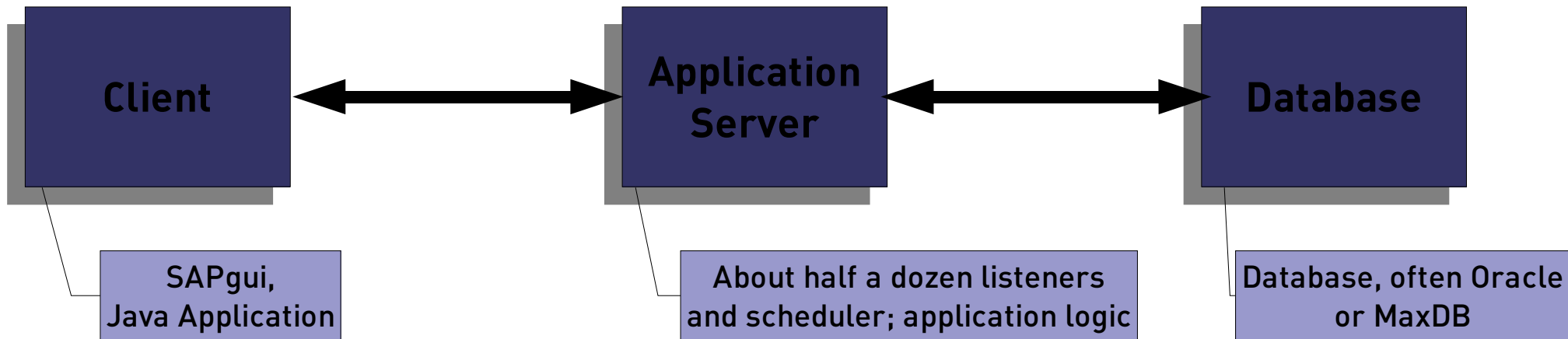
Sometimes difficult to understand SAP people or documentation

SAP makes a great deal of naming everything differently (DIAG, RFC, SAP-routers , ...)

The main achievement seems to be scalability

Simple SAP R/3 setup

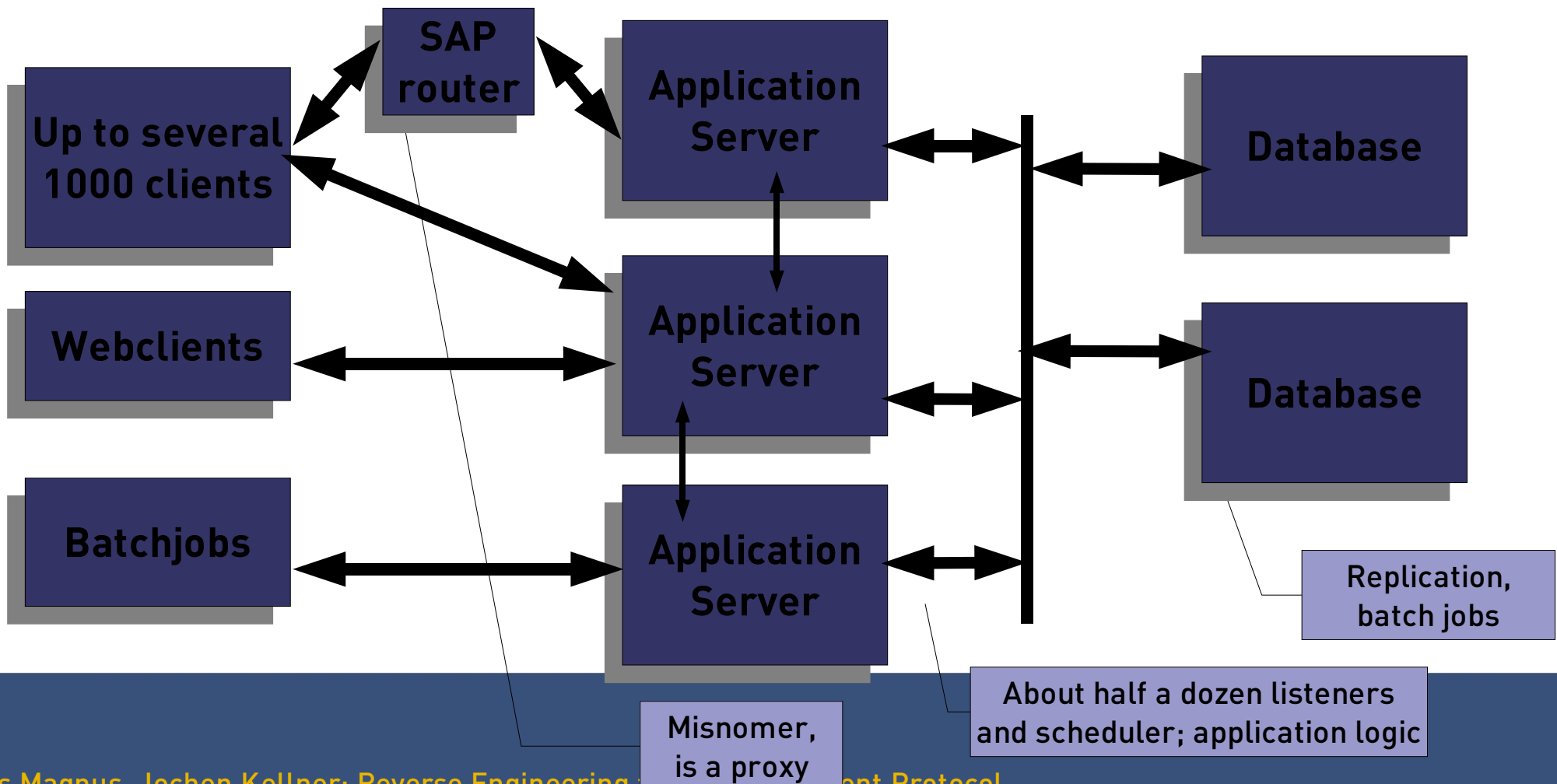
Old fashioned three tier database application



- **Runs on a number of platforms**
- **Supports mainframes, Linux and even Windows**
- **Encapsulates most of the platform**

Complex SAP R/3 setup

Old fashioned three tier database application



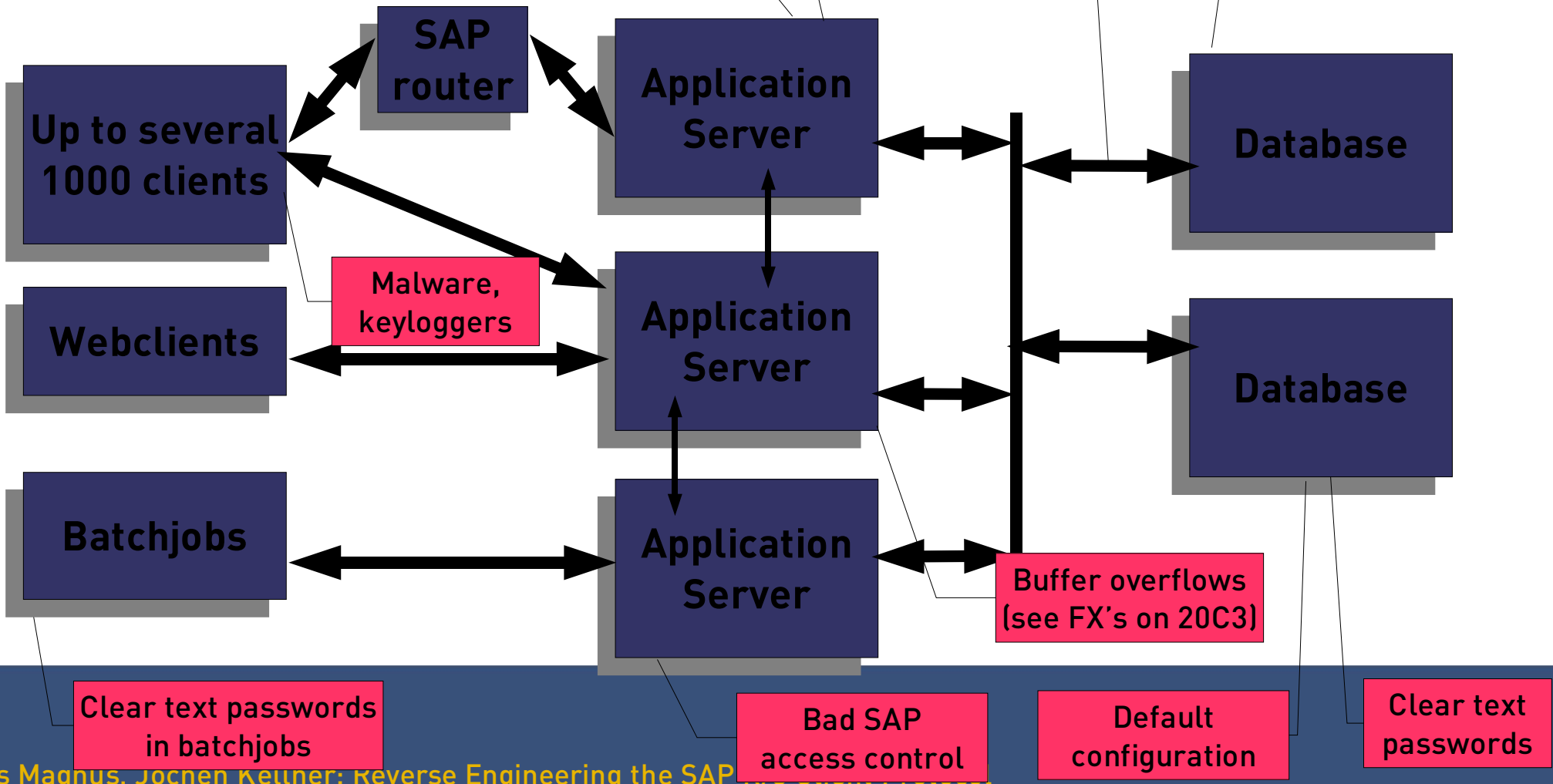
Attacks on SAP

Access to host systems

Unhardened systems

unencrypted protocols

Access to host system



Attacks on SAP installations

Most SAP experts focus solely on application layer issues

- User privileges, access control**

System administrators don't touch SAP

Bad protection on OS level

Important: That's not necessarily SAP's fault

But: What do they do to help it?

Security provided by SAP

A lot of documentation

- Often incomprehensible for networkers

A number of documented APIs

- Plug-in encryption
- Access control

A set of recommendations

- Often not obeyed to by op staff

How to implement security

Allocate lots of time

Understand the system and the language

Harden every server

Place firewalls

Encrypt data transmission

SAP client protocol

Most attacks are commodity attacks that apply to every system

Vulnerabilities to application server have been addressed by FX

Client protocol between sapGUIs and application servers is often unprotected

Once claimed encrypted, now officially disguised

Client protocol details

Protocol internally called `DIAG`

- (not to be confused with the RFC protocol of the same name!)

Full specifications available only with NDA

Stream based network connections

- TCP, but potentially over several other protocols, too

Some details are available within the SAP help

More details

TCP/3200 + x

where x is the instance identifier

C/S-based protocol, exchanging blobs

- 10 Request to AS**
- 20 Response with form data and result data**
- 30 New data and new requests**
- 40 GOTO 20**

Scanner result

```
# nmap (V. 3.00) scan initiated as: nmap -sT -v -p3200-3900 -o nmap-tcp:03.txt 10.36.14.144
```

Interesting ports on (10.36.14.144):

(The 694 ports scanned but not shown below are in state: closed)

Port	State	Service
3200/tcp	open	unknown
3300/tcp	open	unknown
3600/tcp	open	unknown
3773/tcp	open	unknown
3777/tcp	open	unknown
3786/tcp	open	unknown
3900/tcp	open	udt_os

```
# Nmap run completed -- 1 IP address (1 host up) scanned in 22 seconds
```

Trace (client side)

sap-conversation.tcp - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: + Expression... Leeren Anwenden

No.	Time	Source	Destination	Protocol	Info
1	15:55:03.778	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
2	15:55:03.778	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1460
3	15:55:03.780	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=1 Ack=0 Win=65535 Len=0
4	15:55:03.781	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [PSH, ACK] Seq=1 Ack=0 Win=65535 Len=266
5	15:55:03.781	10.36.14.205	10.36.14.144	TCP	[TCP Dup ACK 3#1] 1460 > 3200 [ACK] Seq=267 Ack=0 Win=65535 Len=0
6	15:55:03.798	10.36.14.205	10.36.14.144	TCP	[TCP Dup ACK 3#2] 1460 > 3200 [ACK] Seq=1 Ack=0 Win=65535 Len=0
7	15:55:03.798	10.36.14.205	10.36.14.144	TCP	[TCP Retransmission] 1460 > 3200 [PSH, ACK] Seq=1 Ack=0 Win=65535 Len=266
8	15:55:03.798	10.36.14.205	10.36.14.144	TCP	[TCP Dup ACK 3#3] 1460 > 3200 [ACK] Seq=267 Ack=0 Win=65535 Len=0
9	15:55:04.340	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=267 Ack=2833 Win=62702 Len=0
10	15:55:04.358	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=267 Ack=2833 Win=62702 Len=0
11	15:55:39.547	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [PSH, ACK] Seq=267 Ack=2833 Win=62702 Len=316
12	15:55:39.558	10.36.14.205	10.36.14.144	TCP	[TCP Retransmission] 1460 > 3200 [PSH, ACK] Seq=267 Ack=2833 Win=62702 Len=316
13	15:55:41.598	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=583 Ack=2907 Win=62628 Len=0
14	15:55:41.598	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=583 Ack=2907 Win=62628 Len=0
15	15:55:41.998	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=583 Ack=2981 Win=62554 Len=0
16	15:55:41.998	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [ACK] Seq=583 Ack=2981 Win=62554 Len=0
17	15:55:42.204	10.36.14.205	10.36.14.144	TCP	1460 > 3200 [PSH, ACK] Seq=583 Ack=3727 Win=61808 Len=55

> Frame 6 (60 bytes on wire, 60 bytes captured)
 > Ethernet II, Src: 00:0b:db:d6:b0:d2, Dst: 08:00:20:b8:e6:d0
 > Internet Protocol, Src Addr: 10.36.14.205 (10.36.14.205), Dst Addr: 10.36.14.144 (10.36.14.144)
 > Transmission Control Protocol, Src Port: 1460 (1460), Dst Port: 3200 (3200), Seq: 1, Ack: 0, Len: 0

Block transmission

First 4 octetts are block length

A number of similiar starting octetts

Scrambled data payload

Starts with 0x1f 0x9d

From /etc/magic:

standard unix compress

0	string	\037\235	compress'd data
>2	byte&0x80	>0	block compressed
>2	byte&0x1f	x	%d bits

Compressed data payload

Looks like the LZC algorithm

Also used in old-fashioned compress (1)

Strings LZ.* can be found in sapGUI binary

Just extracting the payload and using uncompress does not work

Bit-length field is wrong

LinuxTag

Leading Free Software and Linux event

Talks and exhibition

Karlsruhe, Germany: June 22 25, 2005

Call for Papers still open until January 15:

<http://www.linuxtag.org/>

Contact

Nils Magnus
Program Chair, LinuxTag e. V.

University of Kaiserslautern
67653 Kaiserslautern
T +49-631-310-9371

magnus@linuxtag.org

