

# **Yakker: A parser generator for network protocol messages**

*Trevor Jim*

*AT&T Labs-Research*

*November 18, 2005*

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

# CAN-2005-1523

**S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)**

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

**C: A001 login trevor foobar**

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

**S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated**

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: **A001** login trevor foobar

S: **A001** OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

C: %n list "" \*

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

C: %n list "" \*

S: . . . .

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

C: %n list "" \*

S: . . . .

```
sprintf(buf, tempbuf, ...);
```

# CAN-2005-1523

S: \* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS AUTH=PLAIN AUTH=LOGIN]  
bigmail.research.att.com IMAP4rev1 2002.336 at Thu, 16 Jun 2005  
13:16:22 -0400 (EDT)

C: A001 login trevor foobar

S: A001 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS  
BINARY SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT  
MULTIAPPEND] User trevor authenticated

C: %n list "" \*

S: ....



```
sprintf(buf, tempbuf, ...);
```

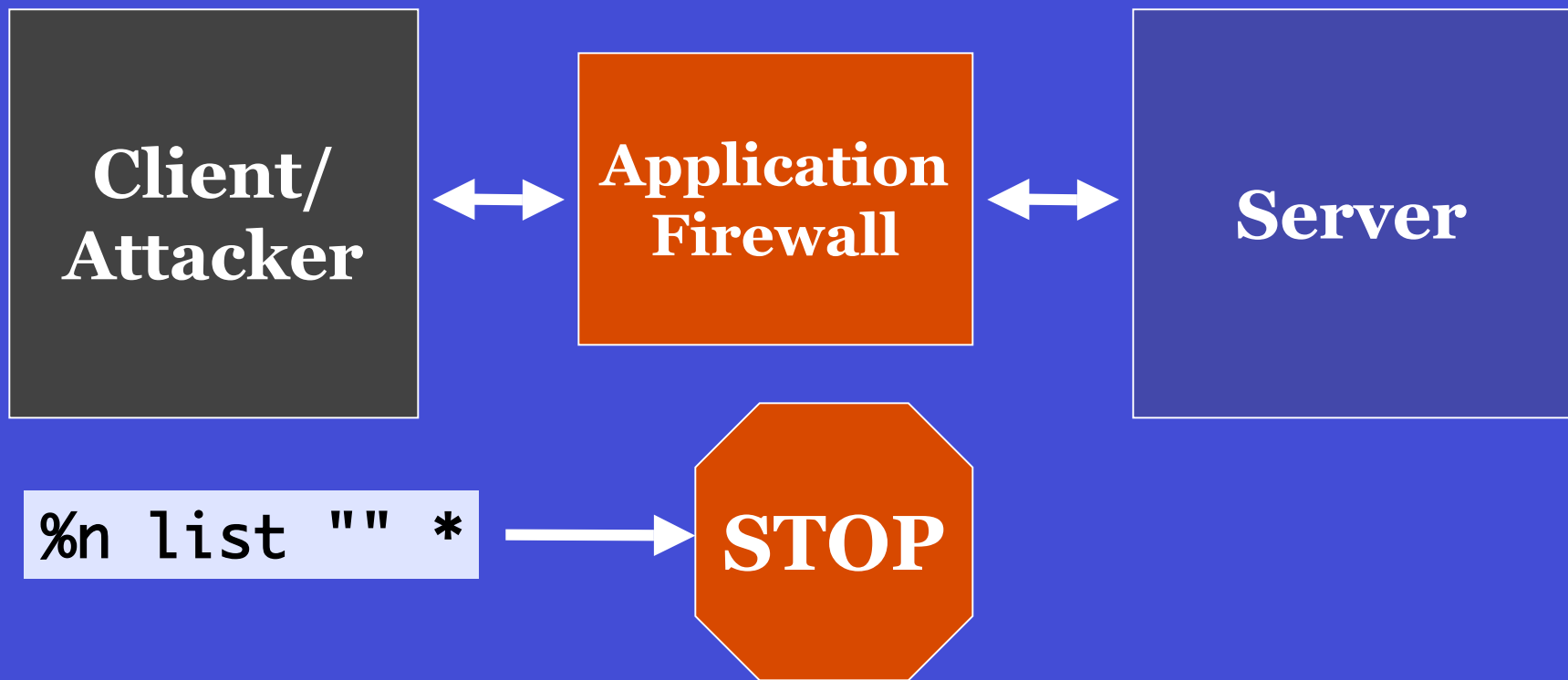
# What can we do?

- Rewrite in a safe language
- Hack the code, fix the bug
- Apply the patch

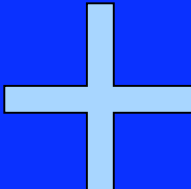
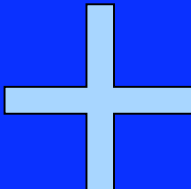
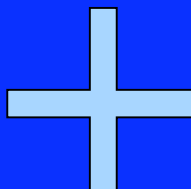
# Coping with malicious messages



# Coping with malicious messages



# Taxonomy of attacks

	Malformed message	Wellformed message
During Parse		
After Parse		

# Taxonomy of attacks

*Malformed /During*

**Ping of death (1996)**

*Wellformed/During*

**Morris Internet Worm (1989)**

*Malformed/After*

**IMAP example (2005)**

# Yakker

- **A parser generator for protocol messages**
- **Goal: build secure servers**
- **Goal: build application firewalls**

# Yakker

- A parser generator for protocol messages
- Goal: build secure servers
- Goal: build application firewalls
- Input is an IETF “Request for Comments” (RFC)

# RFC 3501 — IMAP

Network Working Group  
Request for Comments: 3501  
Obsoletes: 2060  
Category: Standards Track

M. Crispin  
University of Washington  
March 2003

INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1

## Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

## Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

## Abstract

The Internet Message Access Protocol, Version 4rev1 (IMAP4rev1), allows a client to access and manipulate electronic mail messages on a server. IMAP4rev1 permits manipulation of mailboxes (remote message folders) in a way that is functionally equivalent to local folders. IMAP4rev1 also provides the capability for an offline client to resynchronize with the server.

**Why?**

**How?**



# ASN.1

```
AttributeValueAssertion ::= SEQUENCE {  
    attributeDesc  AttributeDescription,  
    assertionValue AssertionValue }
```

```
AssertionValue ::= OCTET STRING
```

```
Attribute ::= SEQUENCE {  
    type  AttributeDescription,  
    vals  SET OF AttributeValue }
```

## LDAP, X.509, . . .

# ABNF

command = tag SP (command-any / command-auth / command-nonauth /  
command-select) CRLF

command-nonauth = login / authenticate / "STARTTLS"

command-select = "CHECK" / "CLOSE" / "EXPUNGE" / copy / fetch / store /  
uid / search

**IMAP, SMTP, “Text messages,” . . .  
“Standardized” in RFC 2234**

# RFC statistics (June 2005)

RFCs total:	3337
RFCs with box:	412
RFCs with ASN.1:	470
RFCs with ABNF:	302

# ***ABNF in Yakker***

# ABNF extraction

- Slice 'n dice
- Parse the slices
- Prose elimination

```
tag = 1*<any ASTRING-CHAR except "+">
```

# Parser generation

- **Standard top-down recursive descent**

command = tag SP (command-auth|command-nonauth) CRLF

```
p_command() {  
    p_tag();  
    p_SP();  
    if (lookahead())  
        p_command_auth();  
    else  
        p_command_nonauth();  
    p_CRLF();  
}
```

# Parsing difficulties

- **No lexer/parser separation**
  - No Lex/Yacc
  - Need  $k$ -token lookahead

"Jan"|"Feb"|"Mar"|"Apr"|"May"|...

- Need unbounded lookahead!

$n = 1^* \text{DIGIT}$

$r = n \text{ ":" } n$

$(n \mid r)$

# Parsing difficulties

- Grammar ambiguous, order does not help

(body-type-basic|body-type-msg|body-type-text)

body-type-basic = media-basic SP ...

media-basic = (string | ...) ...

body-type-msg = media-message SP ...

media-message = DQUOTE "MESSAGE" DQUOTE ...

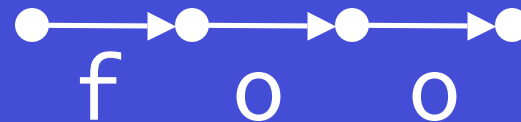
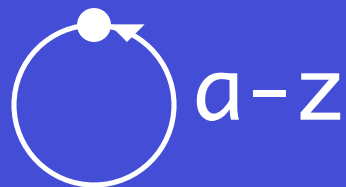
body-type-text = media-text SP ...

media-text = DQUOTE "TEXT" DQUOTE ...

# Lookahead

(body-type-basic|body-type-msg|body-type-text)

- **Build a DFA for each alternative**
- **Merge DFAs**
- **Break ties by probability**



# An IMAP parser

Network Working Group  
Request for Comments: 3501  
Obsoletes: 2060  
Category: Standards Track

M. Crispin  
University of Washington  
March 2003

INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1

+

literal = "{" number  $\$x$  "}" CRLF @repeat(atoi(x)) CHAR8

=

**43 LL(1) conflicts  
in 157 definitions**

# An IMAP Firewall (1 of 2)

IMAP parser  
+

```
my-command = command$x { write(servfd,x,numelts(x)-1); }
```

```
my-greeting = greeting$x { write(clifd,x,numelts(x)-1); }
```

```
my-response = response$x { write(clifd,x,numelts(x)-1); }
```

# An IMAP Firewall (2 of 2)

```
p_my_greeting(serv);  
  
if (fork() == 0)  
    for (;;) p_my_command(cli);  
  
else  
    for (;;) p_my_response(serv);
```

**Future work**

# More protocols

- **Bigger grammars: HTTP, SIP (301 rules)**
  - **Problem: implicit whitespace**
- **Box diagrams**
- **ASN.1**

# Correctness

- **Why is my approach better?**
  - **Multiple back ends**
  - **Build test cases automatically**
  - **Generate proof of safety or correctness**

# Summary

- **Yakker constructs parsers from RFCs**
- **Easily build simple application firewalls**
- **Helps build more secure servers**
- **Many more uses...**