A Stateful Inspection of FireWall-1

data protect

Thomas Lopatic, John McDonald
TÜV data protect GmbH

tl@dataprotect.com, jm@dataprotect.com

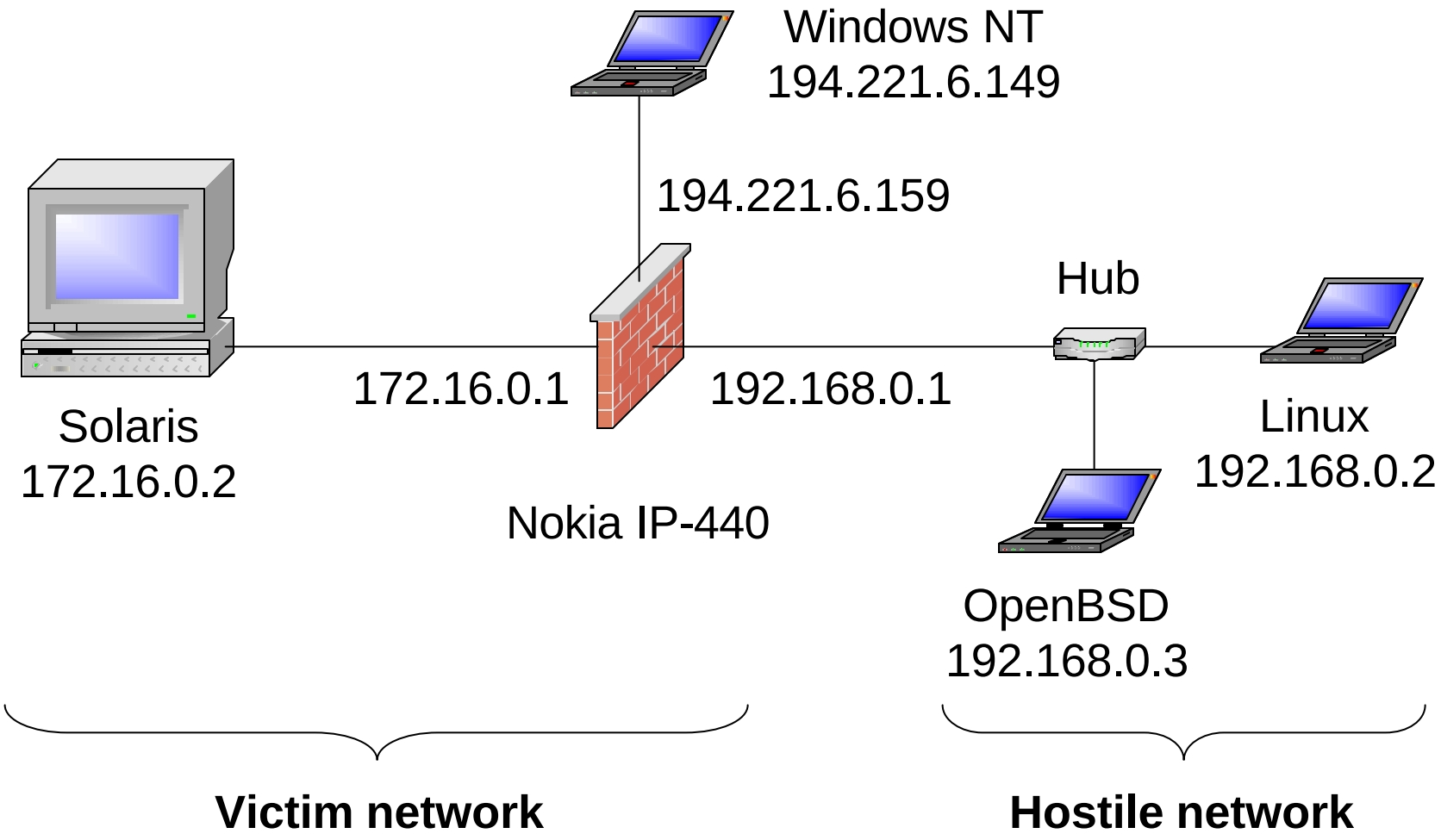
Dug Song
CITI at the University of Michigan

dugsong@umich.edu

Overview

- Architecture of FireWall-1
- Attacking the firewall’s state I
- FWZ encapsulation
- Attacking the firewall’s state II
- Attacking authentication between firewall modules
- Hardening FireWall-1
- The big picture

Topology


Windows NT
194.221.6.149
194.221.6.159
Hub
Solaris
172.16.0.2
172.16.0.1
192.168.0.1
Linux
192.168.0.2
Nokia IP-440
OpenBSD
192.168.0.3
Victim network
Hostile network

Problems in Inspection

- Unreliable / unauthenticated input
- Layering restrictions on inspection
- Layering violations in inspection
- Ambiguous end-to-end semantics

Example: Airport Security

- Unreliable / unauthenticated input

  Examining baggage tags

- Layering restrictions on inspection

  Examining shape, size, weight

- Layering violations in inspection

  Parallelizing bag content inspection

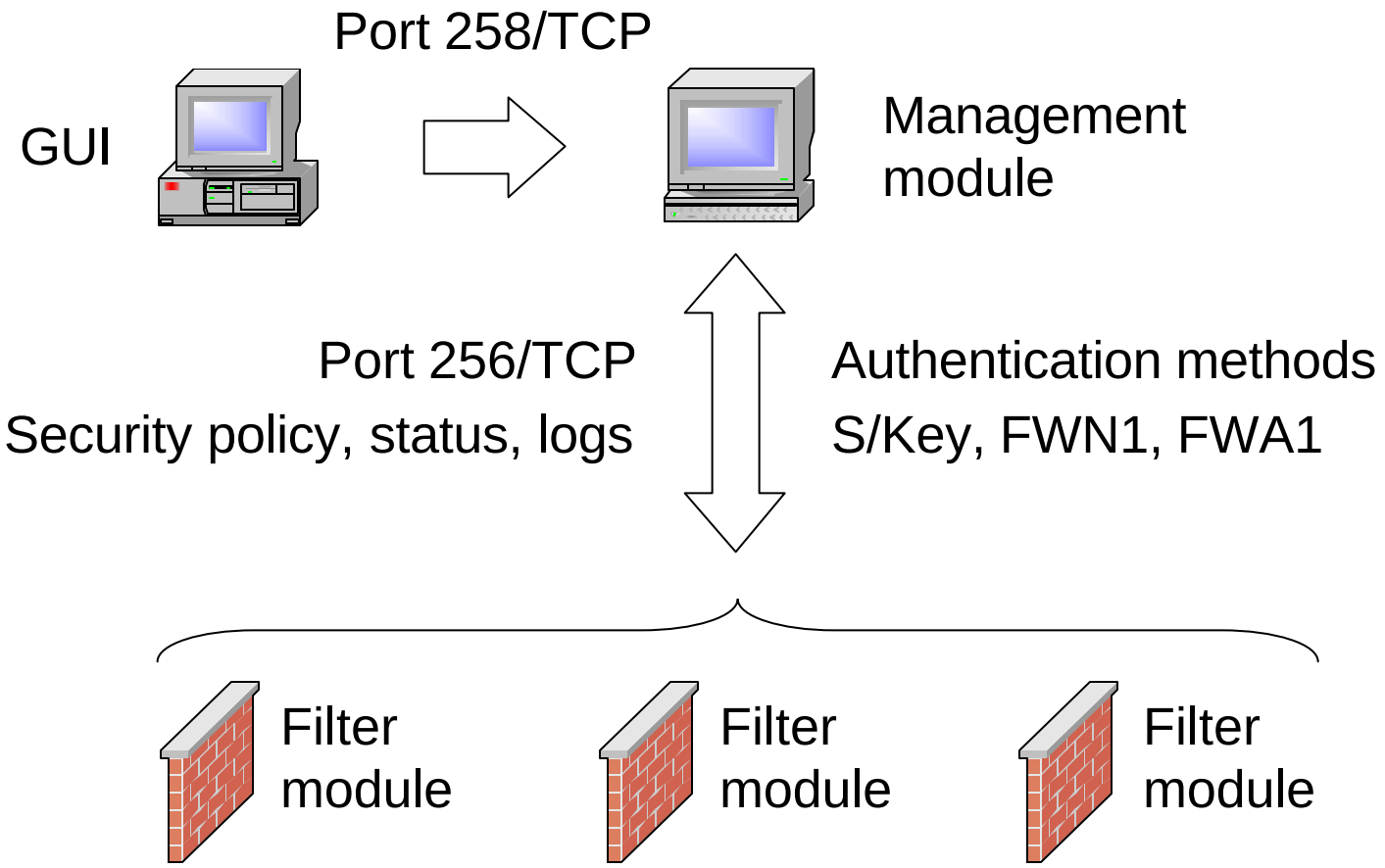
- Ambiguous end-to-end semantics

  Checking for known contraband

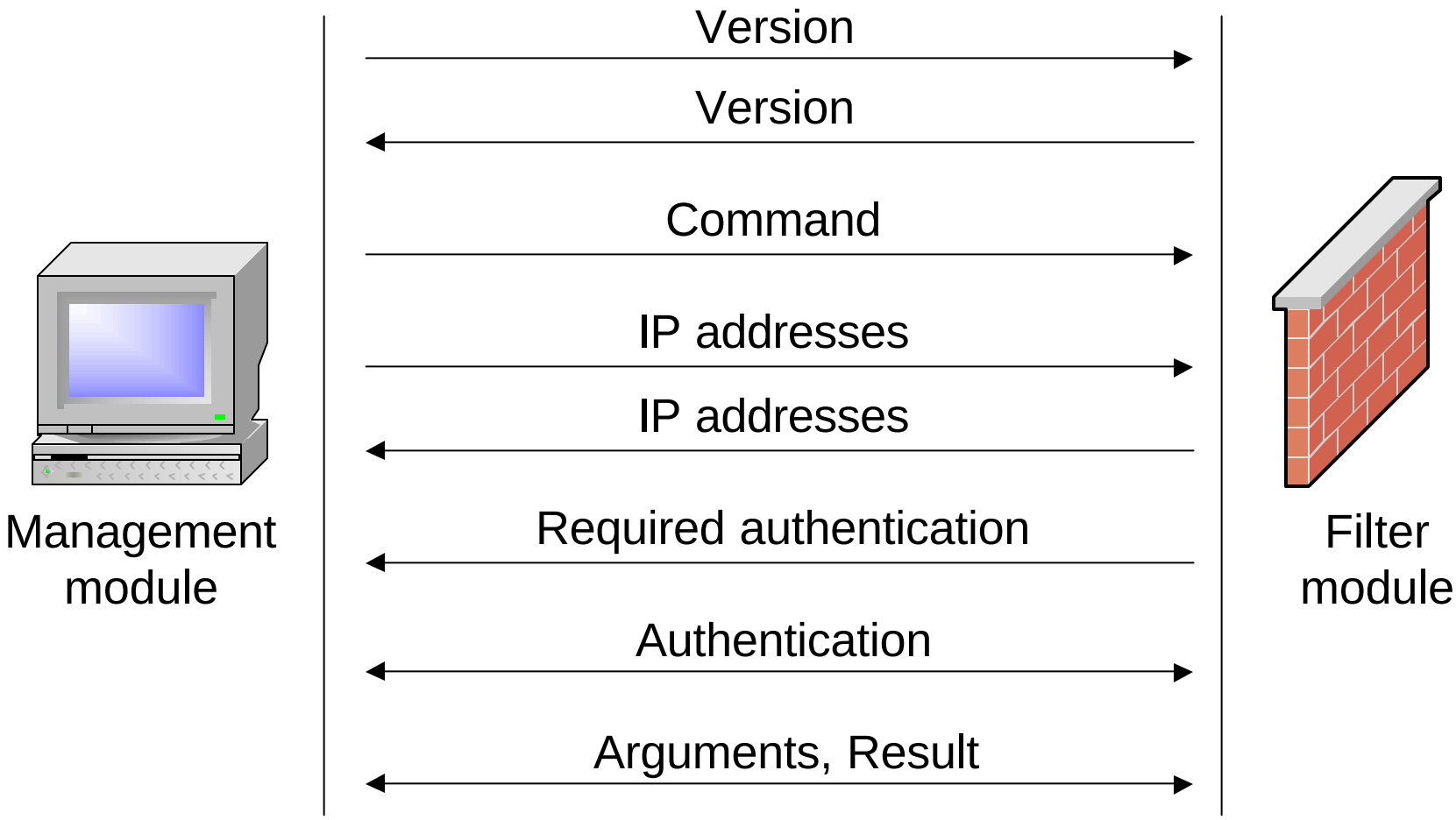
Classification of the Attacks

- Unreliable / unauthenticated input
  - **TCP fastmode**
- Layering restrictions on inspection
  - **FWZ VPN encapsulation**
- Layering violations in inspection
  - **FTP data connection handling**
  - **unidirectional TCP data flow**
  - **RSH error connection handling**
- Ambiguous end-to-end semantics
  - **Parsing of FTP “PORT” commands**

FireWall-1 Modules

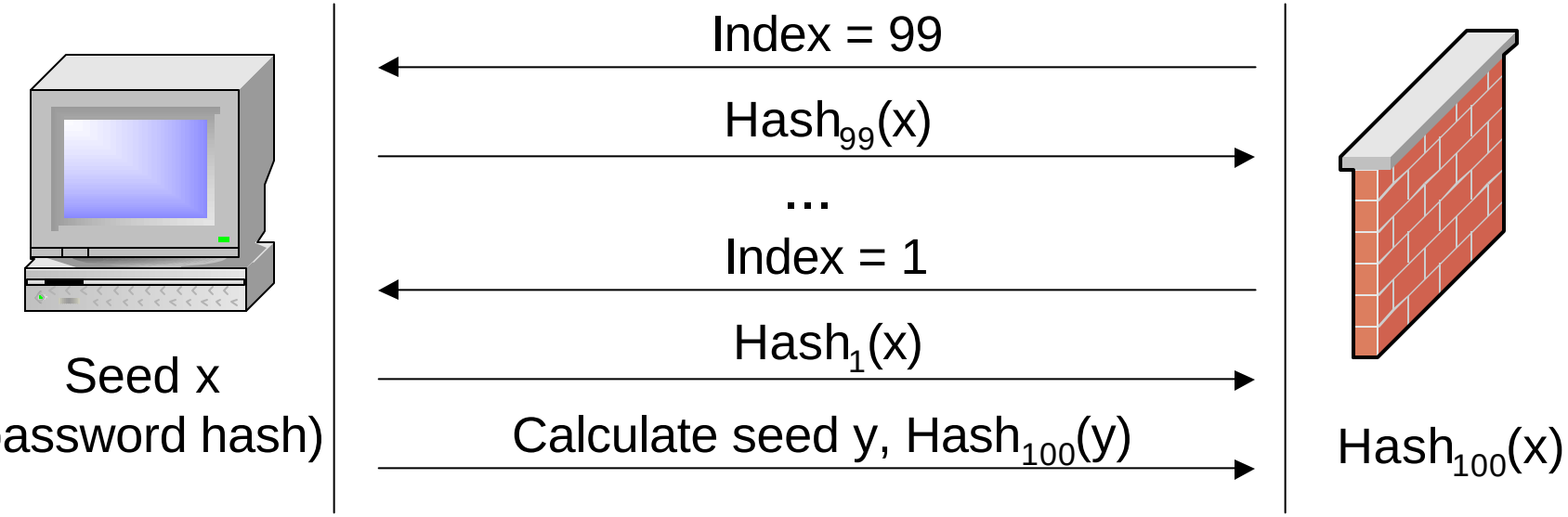


Inter-Module Protocol



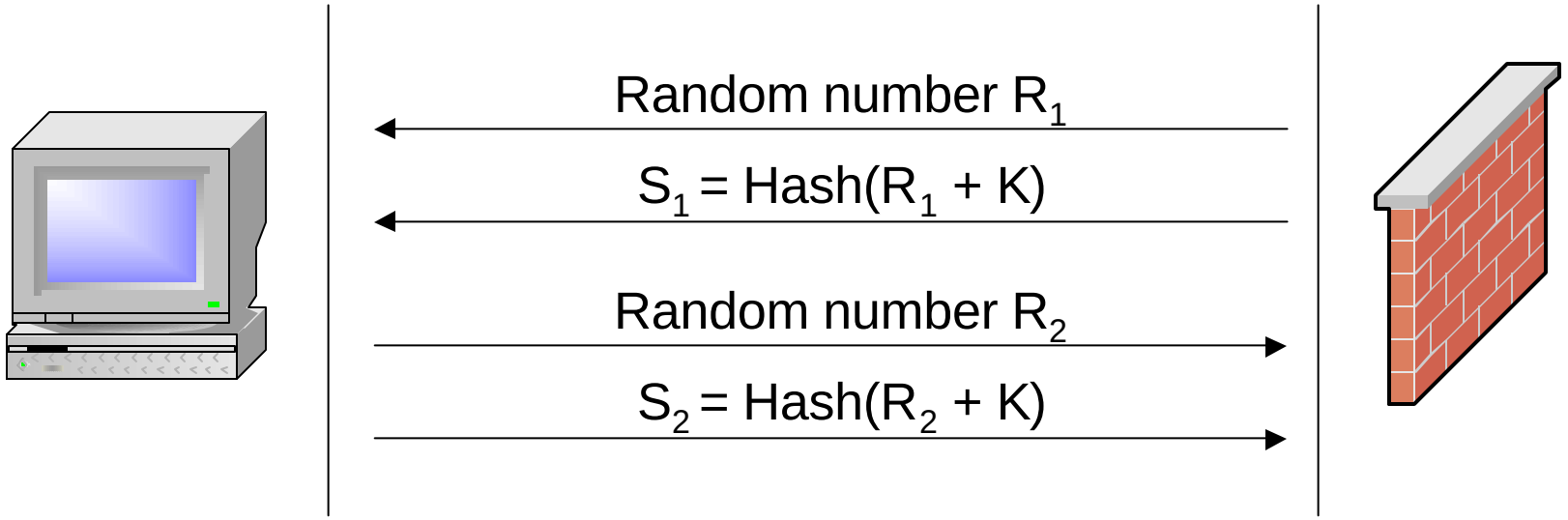
S/Key Authentication

$$Hash_n(x) = \underbrace{Hash(Hash(\ldots Hash(x)))}_{n \text{ times}} = Hash(Hash_{n-1}(x))$$



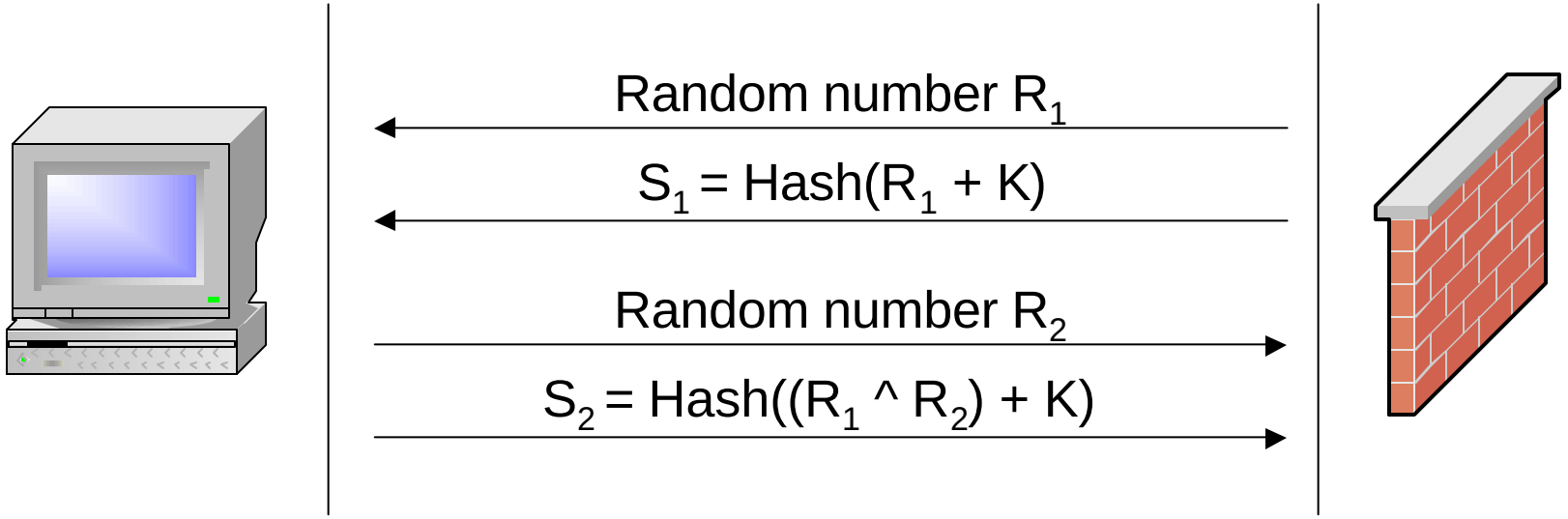
- "y = MakeSeed(time(NULL))"
- Attack: brute force

FWN1 Authentication



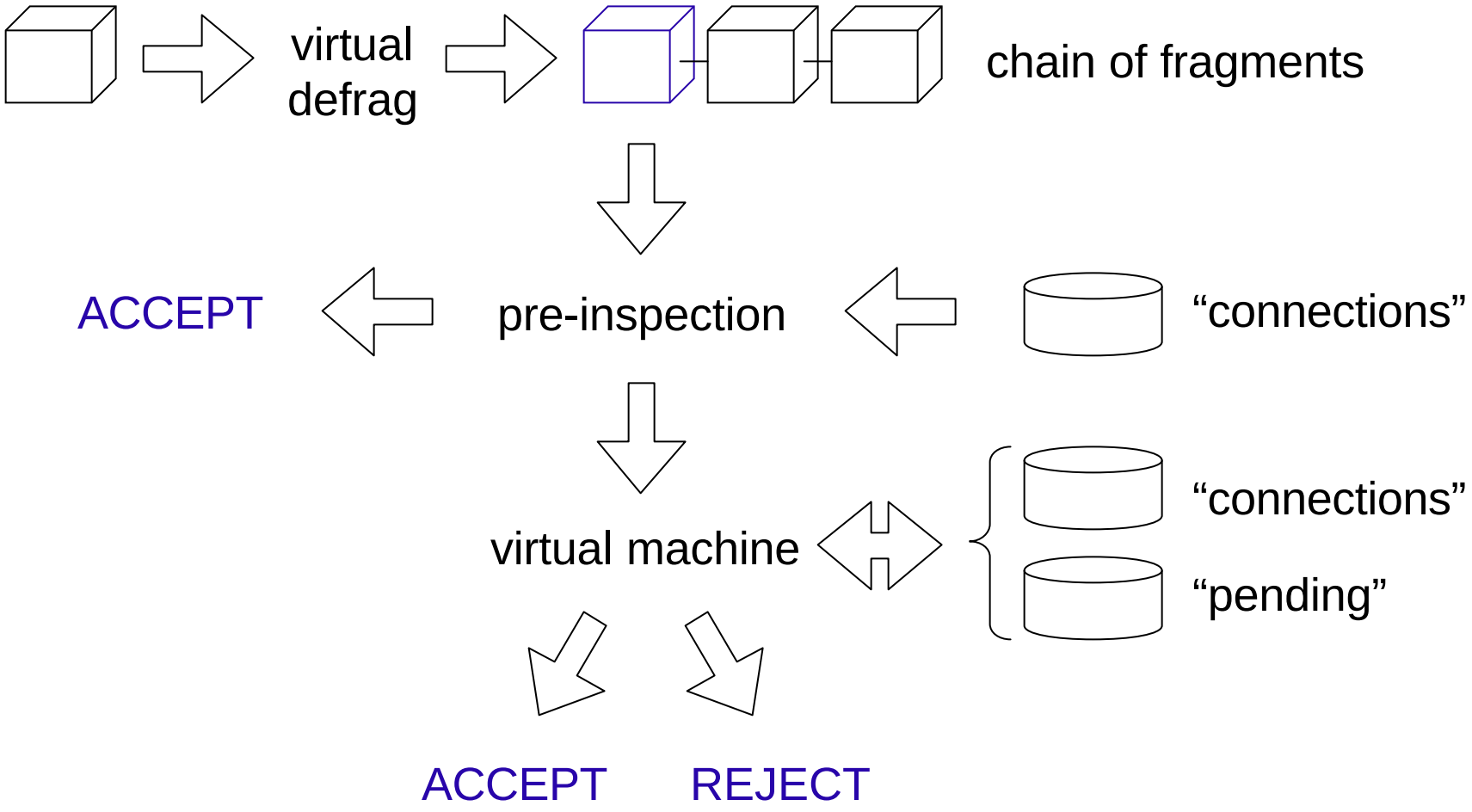
- Shared key *K* (“fw putkey”)
- Attack: choose $R_2 = R_1$, so that $S_2 = S_1$

FWA1 Authentication

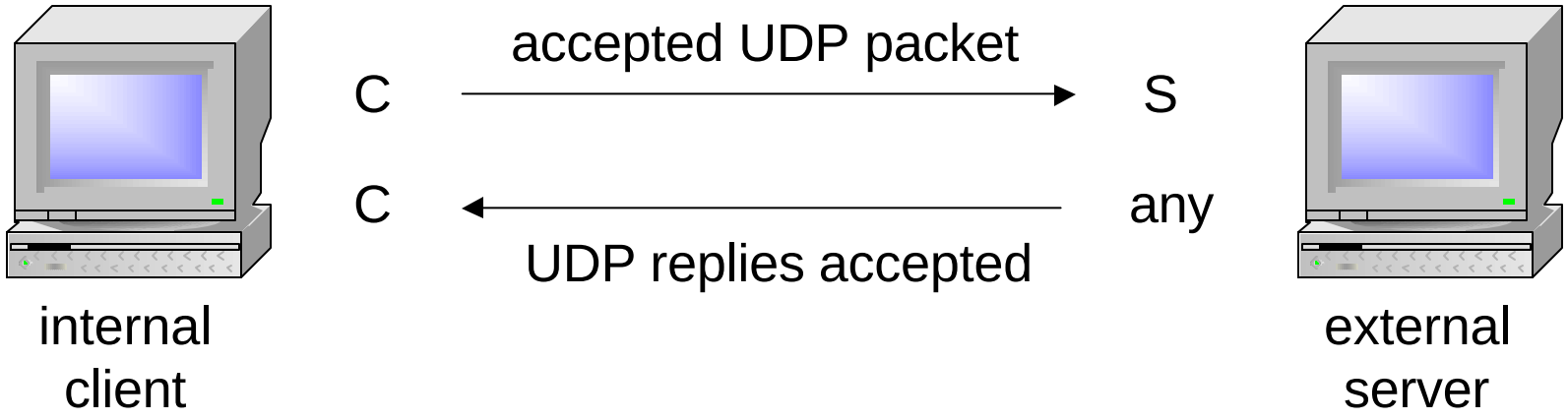


- Shared key *K* (“fw putkey”)
- Attack: choose R_2 = 0, so that
  - R_1 ^ R_2 = R_1 and
  - S_2 = Hash((R_1 ^ R_2) + K) = Hash(R_1 + K) = S_1
- To be solved: encryption

Stateful Inspection I

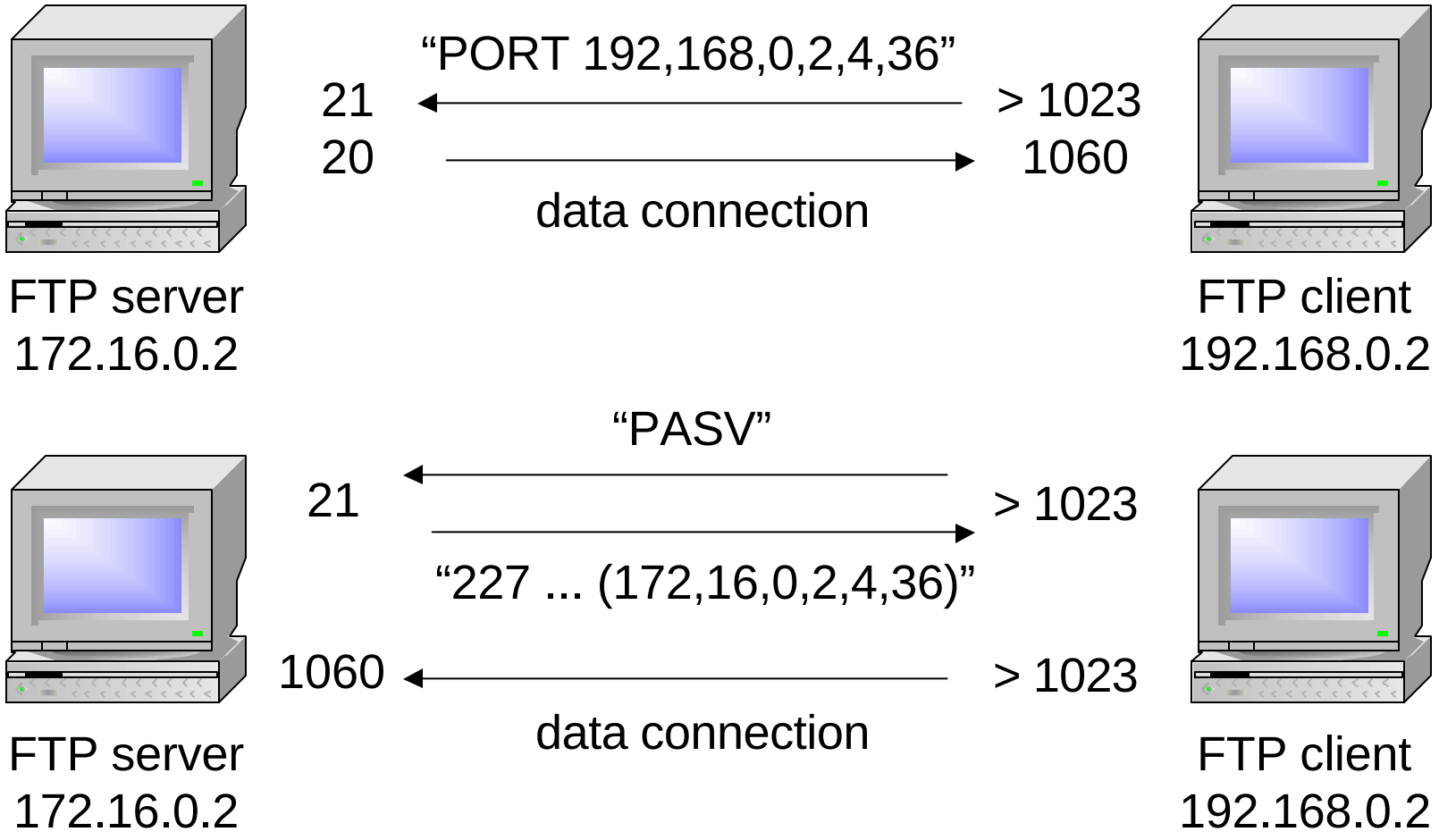


Stateful Inspection II

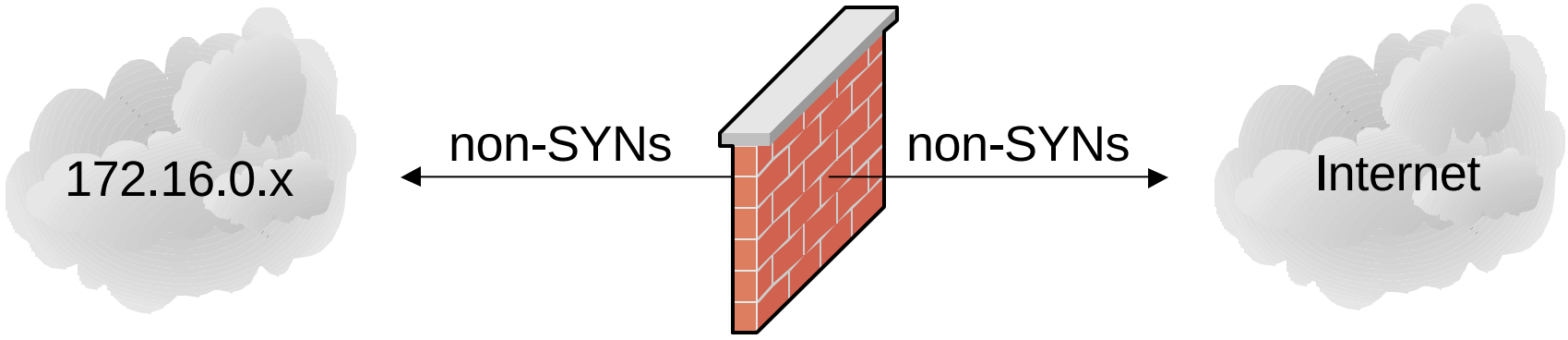


- UDP “connections”
  - from a client, port C
  - to a server, port S + wildcard port
- <s-address, s-port, d-address, d-port, protocol>

Stateful Inspection III

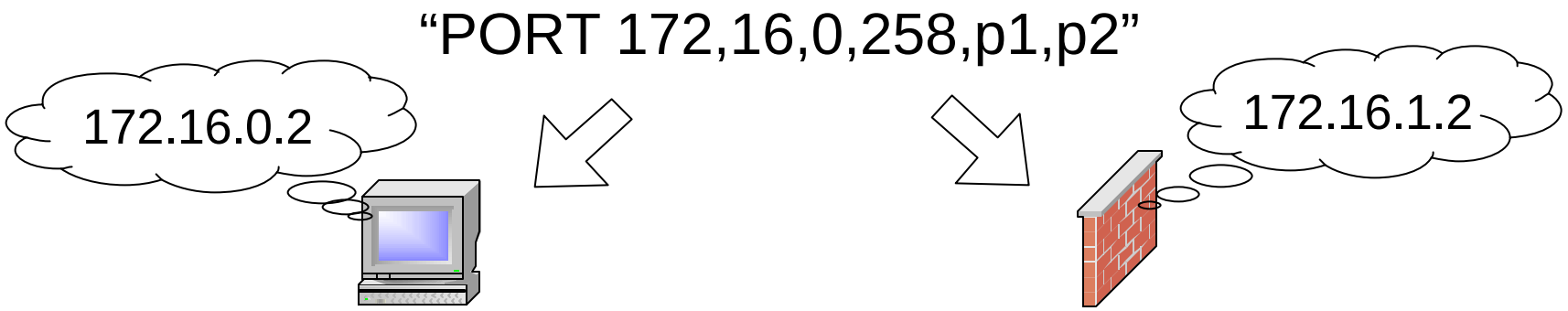


Fastmode Services



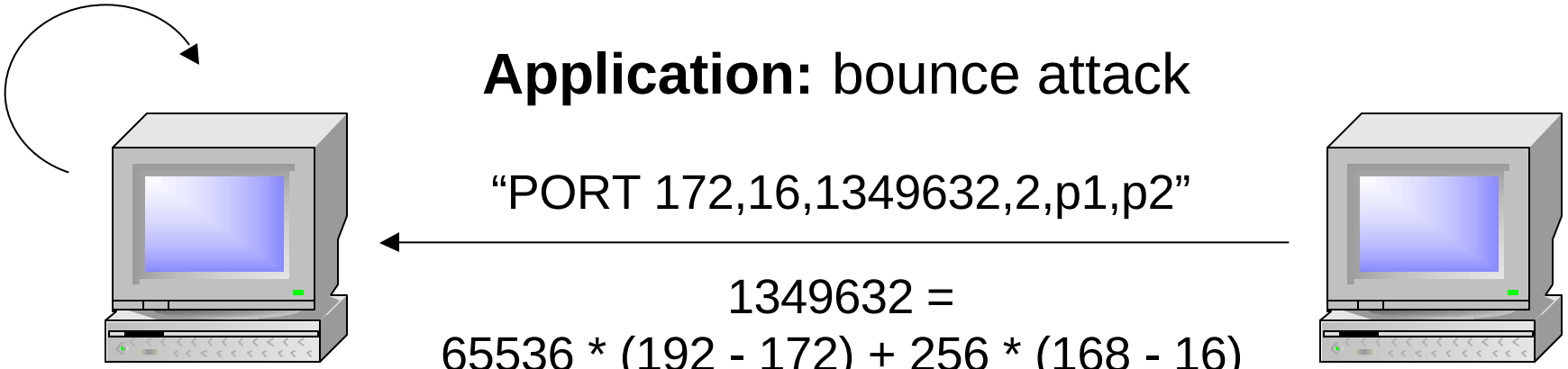
- non-SYN packets accepted
  - Source port = fastmode service
  - Destination port = fastmode service
- Stealth scanning (FINs, ...)

FTP “PORT” Parsing



data connection

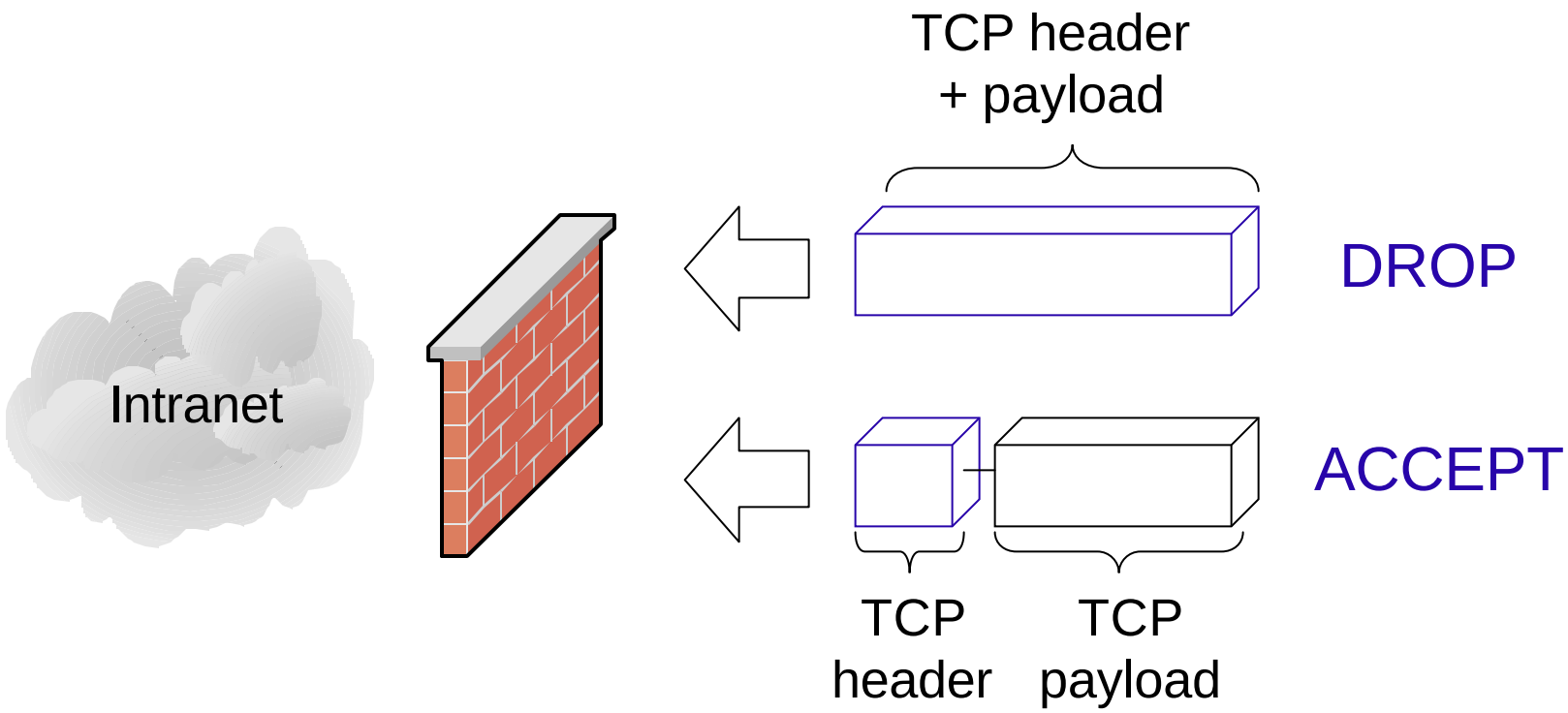
Application: bounce attack



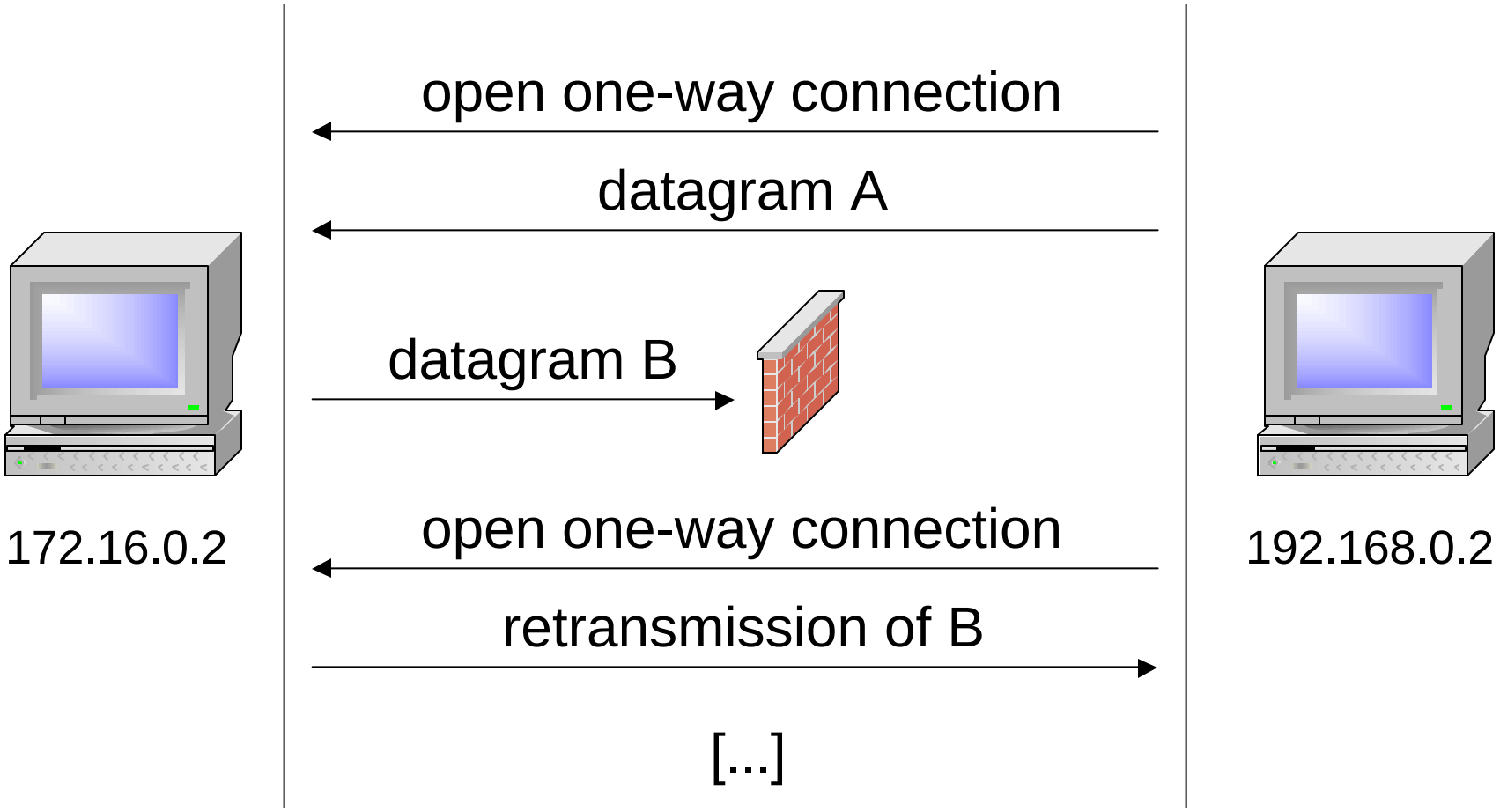
FTP “PASV” Handling

- Advertise small Maximal Segment Size
- Server replies split

One-way Connections I

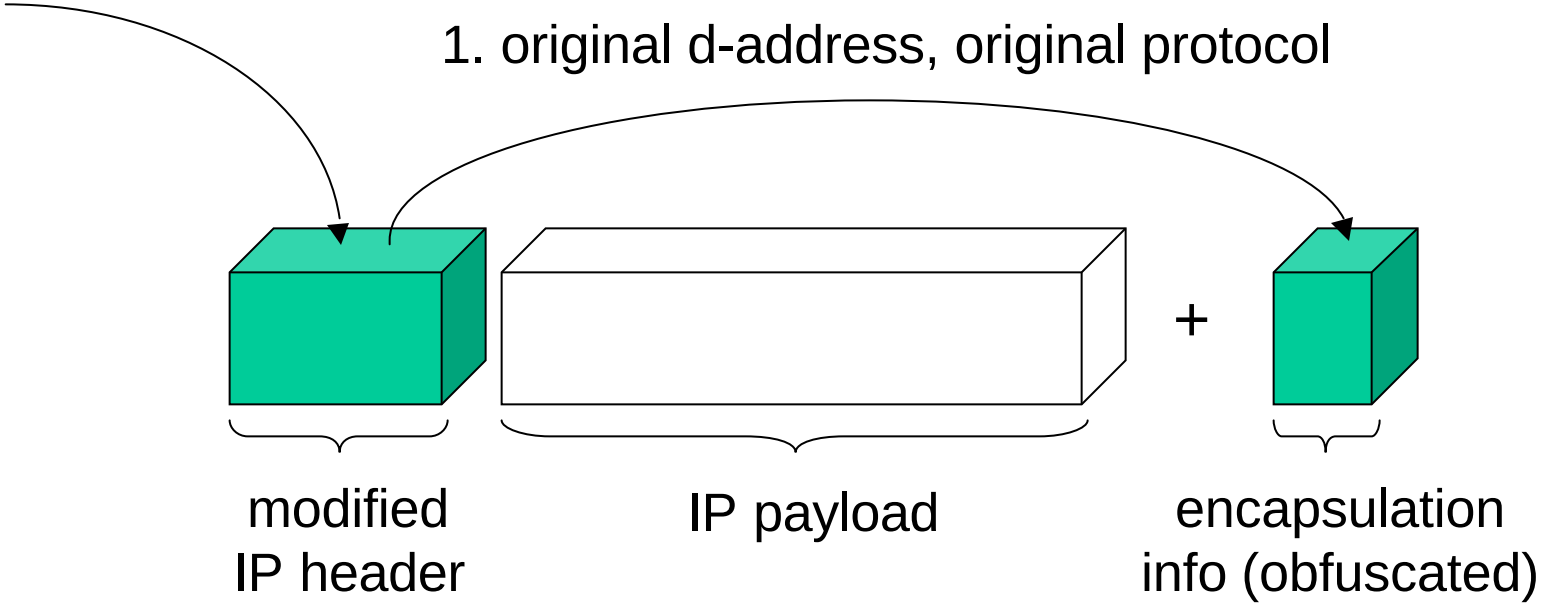


One-way Connections II



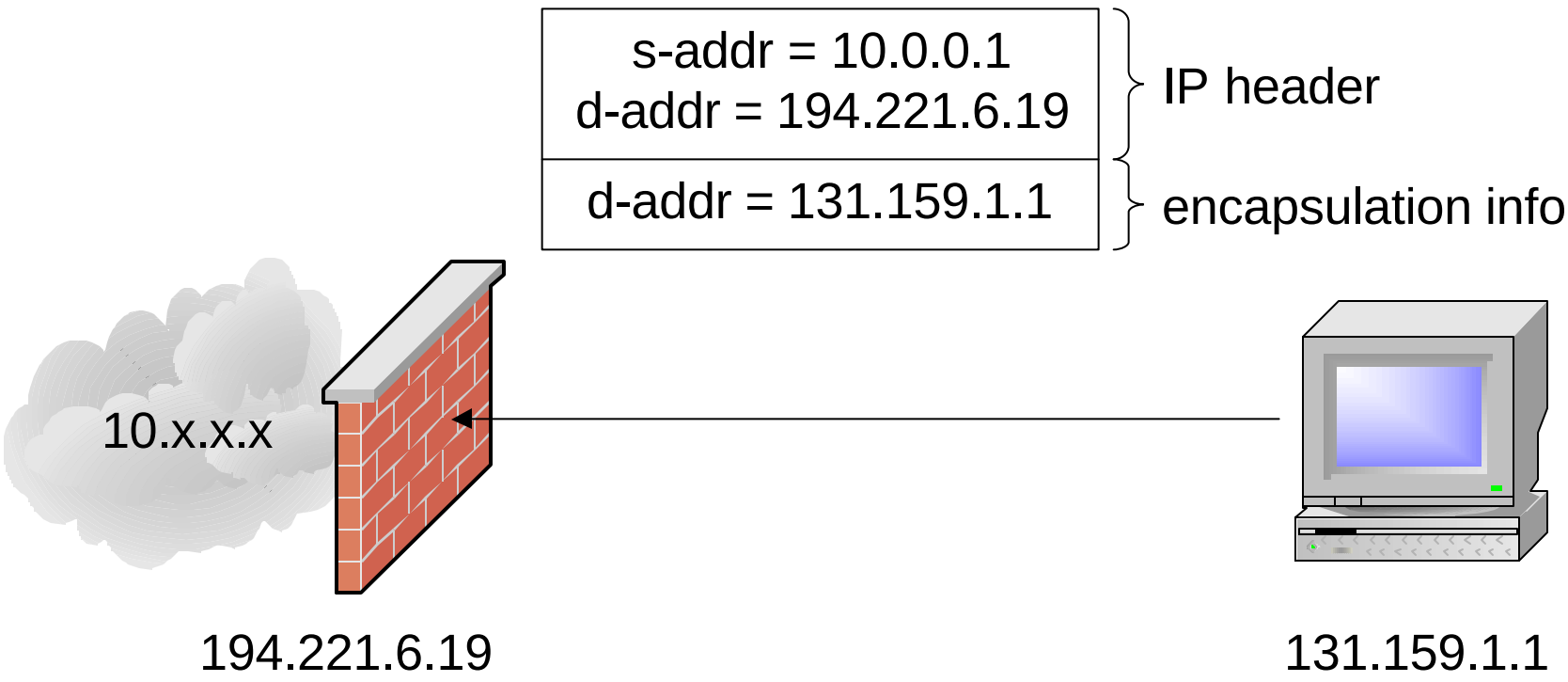
FWZ Encapsulation I

2. d-address = firewall, protocol = 94



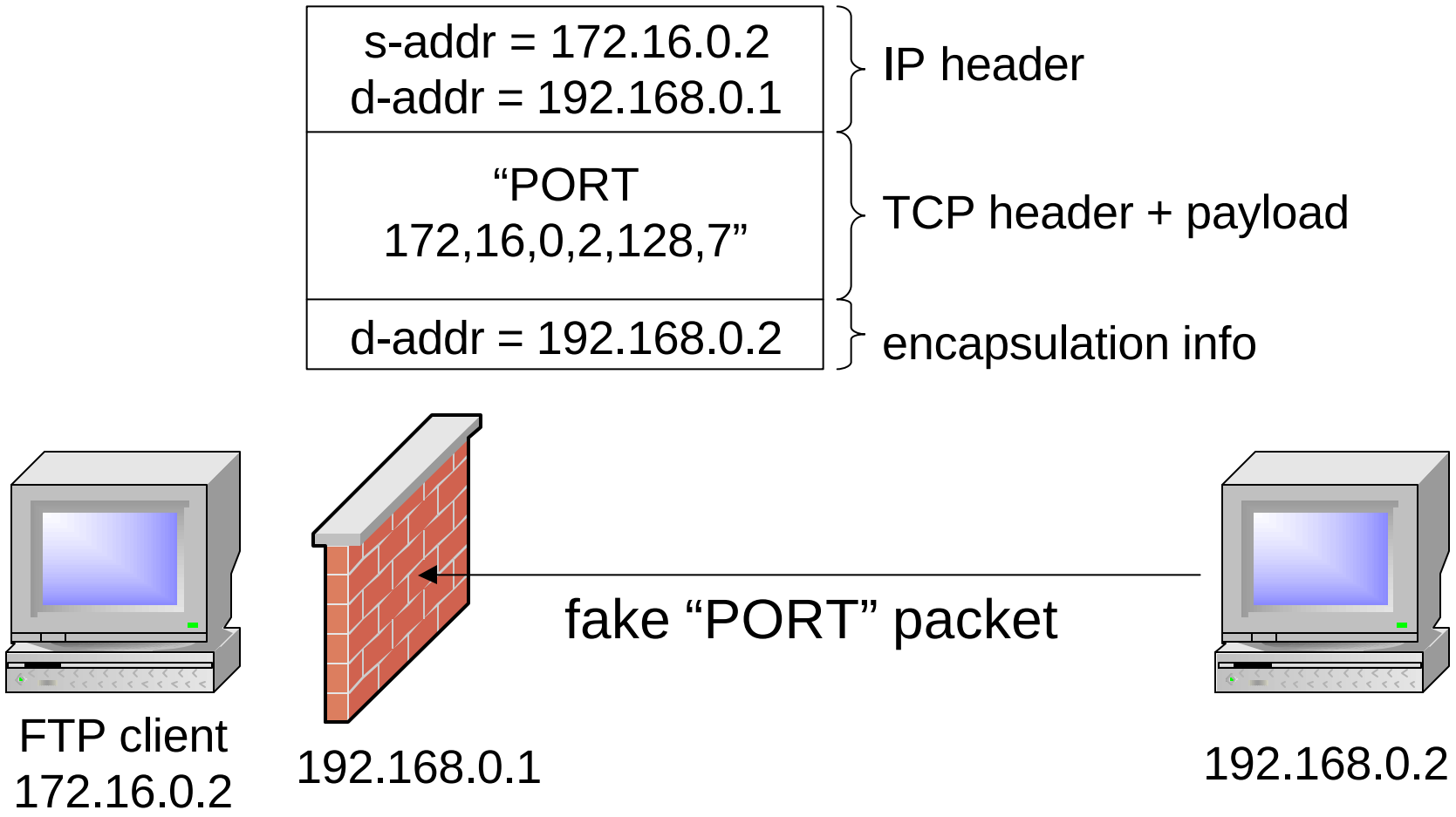
- VPN tunneling protocol
- Decapsulation without decryption or authentication
- Cannot be disabled

FWZ Encapsulation II

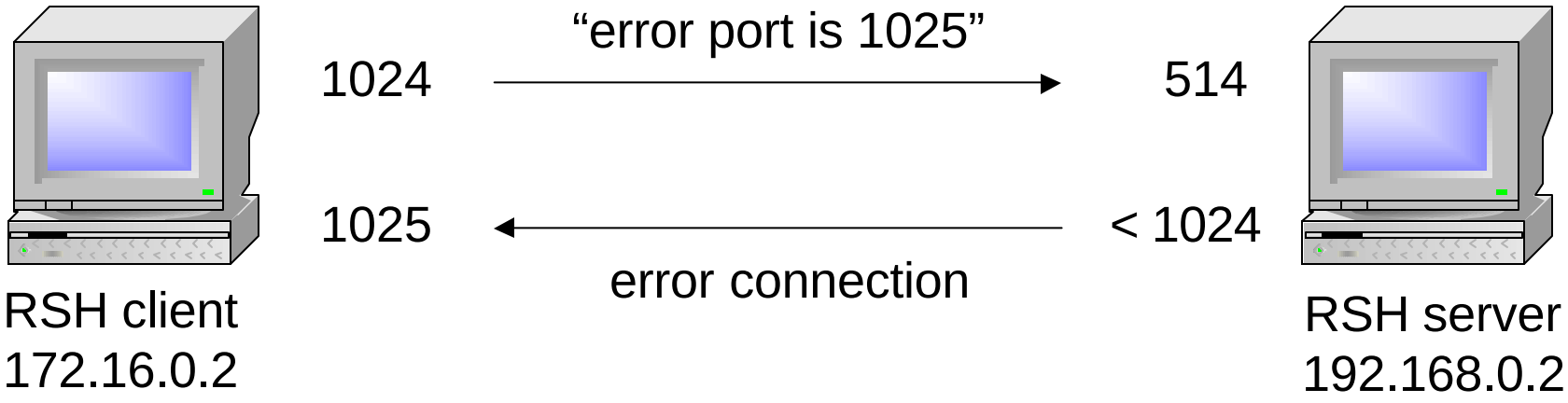


Key to spoofing attacks

Fake “PORT” Commands

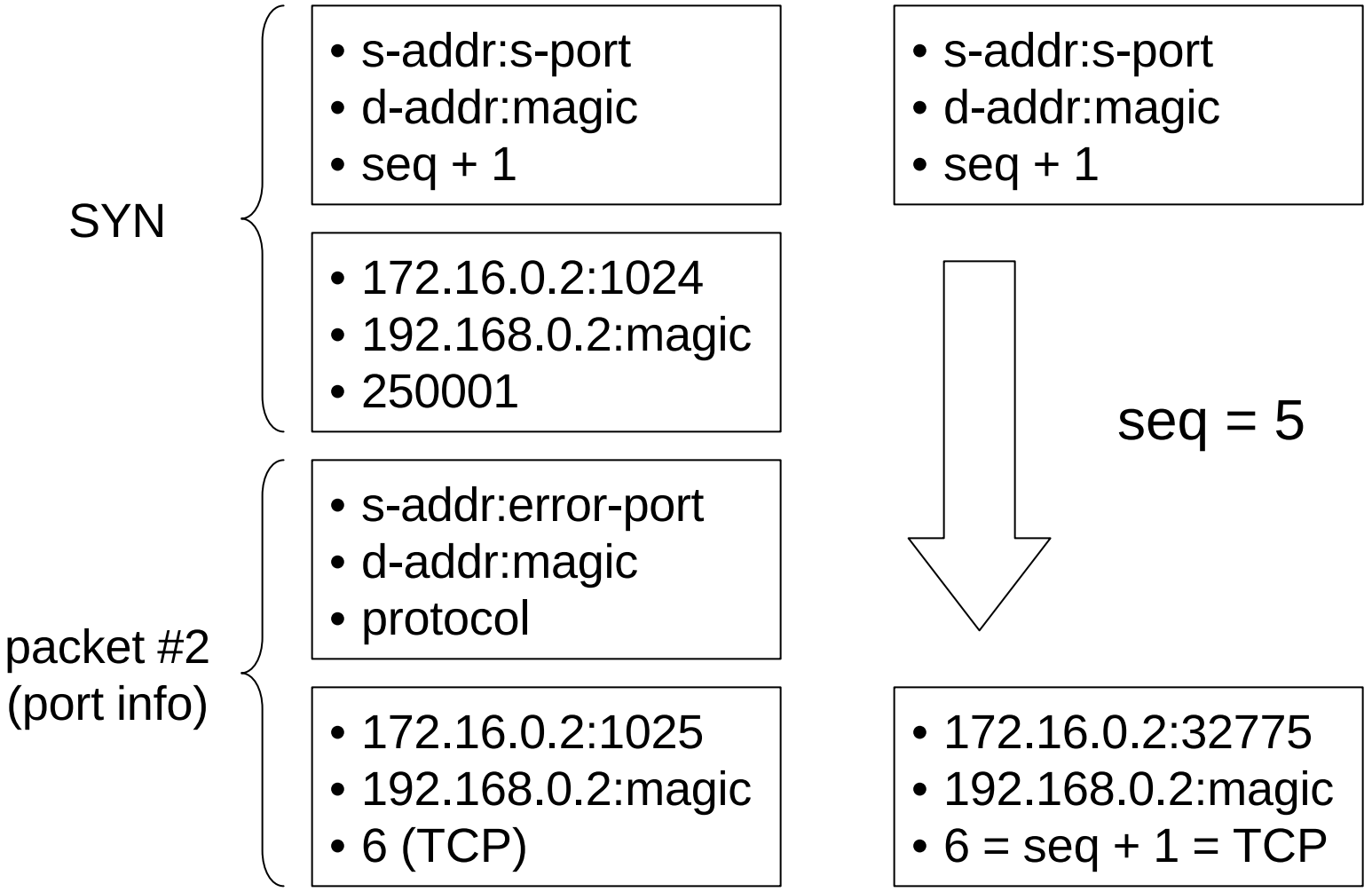


RSH Error Connections I

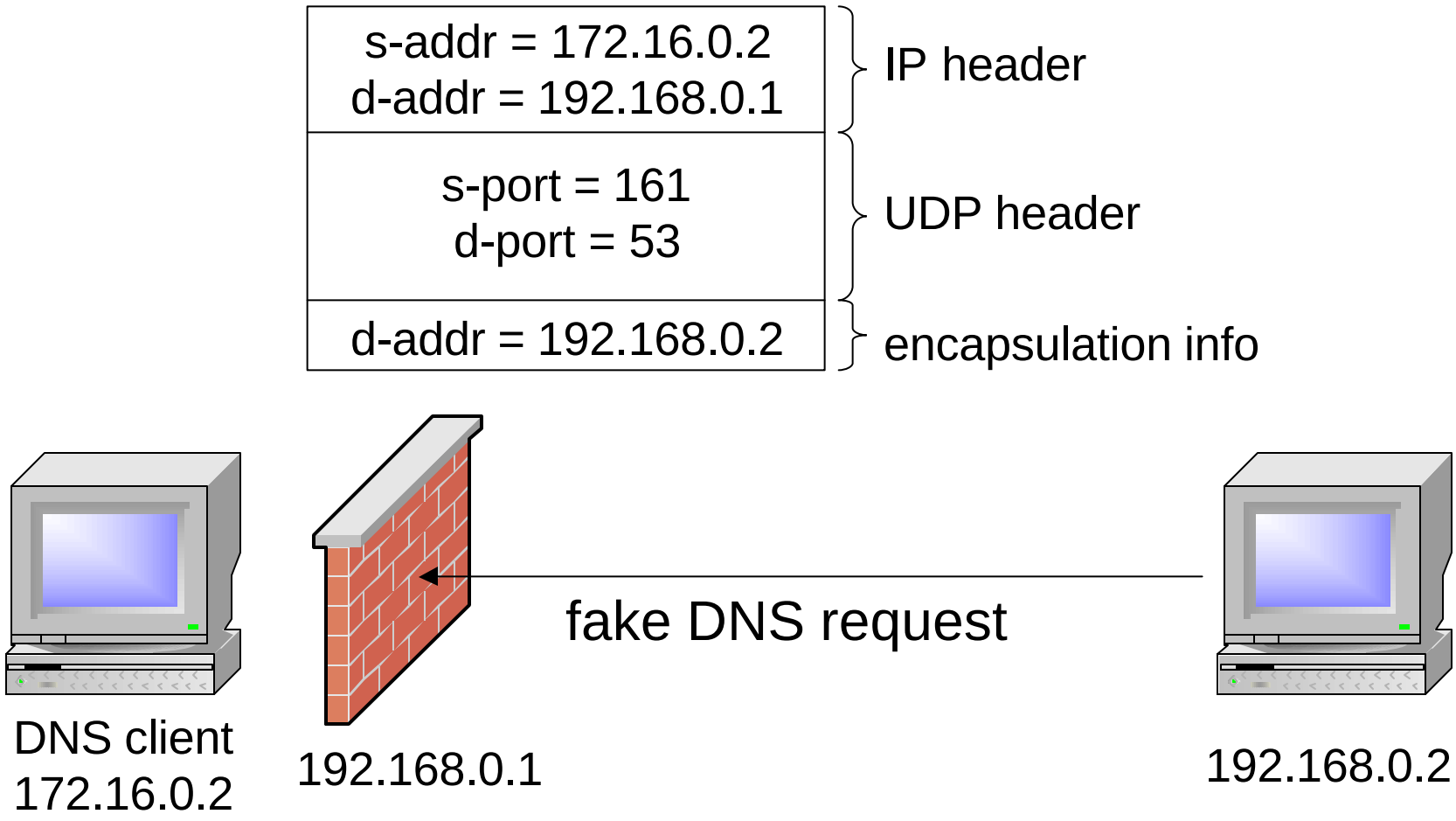


- <172.16.0.2, 1024, 192.168.0.2, 514, 6> in “connections”
- <172.16.0.2, 1025, 192.168.0.2, magic, 6> in “pending”
- Reversed matching

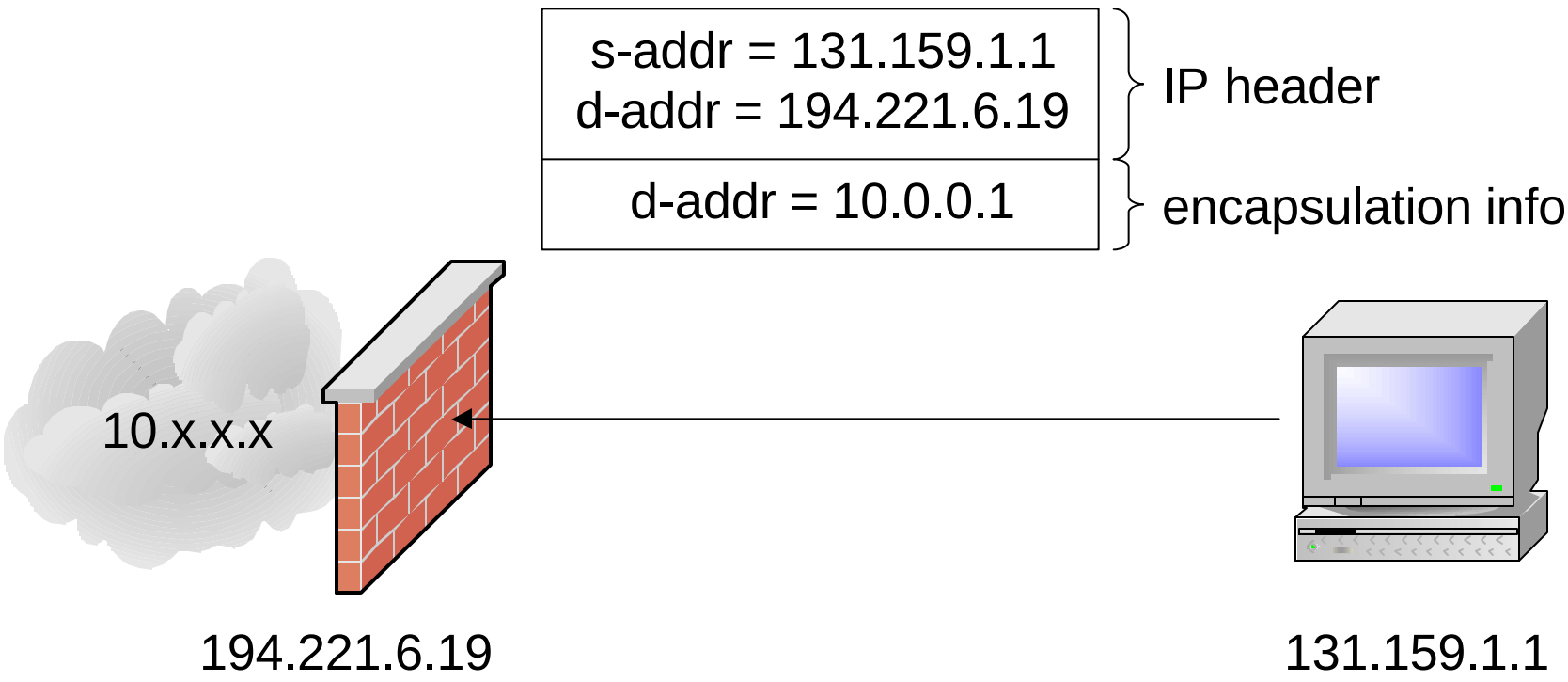
RSH Error Connections II



Fake UDP Requests



FWZ Encapsulation III



Key to non-routable addresses

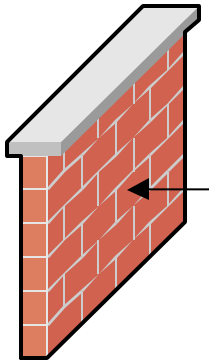
Anti-Spoofing Protection I

1.

s-addr = 192.168.0.1 d-addr = 192.168.0.1
s-port = 161 d-port = 53
d-addr = 192.168.0.2

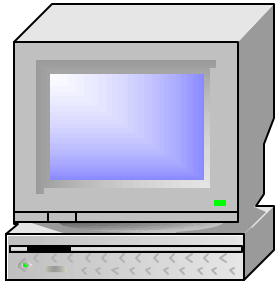
2.

s-addr = 192.168.0.2 d-addr = 192.168.0.1
s-port = any d-port = 161



1. fake DNS request

2. tunnel to firewall



192.168.0.1

192.168.0.2

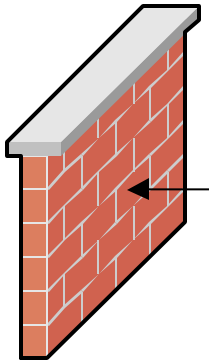
Anti-Spoofing Protection II

1.

s-addr = 224.0.0.1 d-addr = 192.168.0.1
s-port = 161 d-port = 53
d-addr = 192.168.0.2

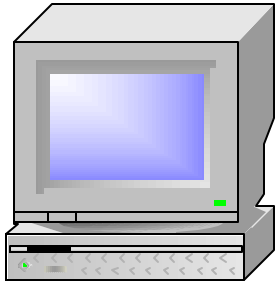
2.

s-addr = 192.168.0.2 d-addr = 192.168.0.1
s-port = 53 d-port = 161
d-addr = 224.0.0.1



1. fake DNS request

2. tunnel to firewall



192.168.0.1

192.168.0.2

Hardening I

- Disable implicit rules
  - DNS
  - control connections
  - ICMP
- Restrictive access rules
  - no “any” sources or destinations
  - deny broadcast / multicast addresses
  - “minimal privilege”
- Properly configure anti-spoofing mechanism
- Filter protocol 94 (e.g. IP Filter)

Hardening II

- Different (virtual) IP addresses for public services
- Restrict control connections
  - FWA1 authentication
  - VPN technology
  - **never** use “127.0.0.1: */none”
- More than one line of defense!

Fixes by Check Point

Solutions by Check Point available today at

http://www.checkpoint.com/techsupport

Thanks.

Thomas Lopatic
tl@dataprotect.com

John McDonald
jm@dataprotect.com

Dug Song
dugsong@umich.edu