Software and Security Engineering

Lecture 6

Anil Madhavapeddy

avsm2@cam.ac.uk

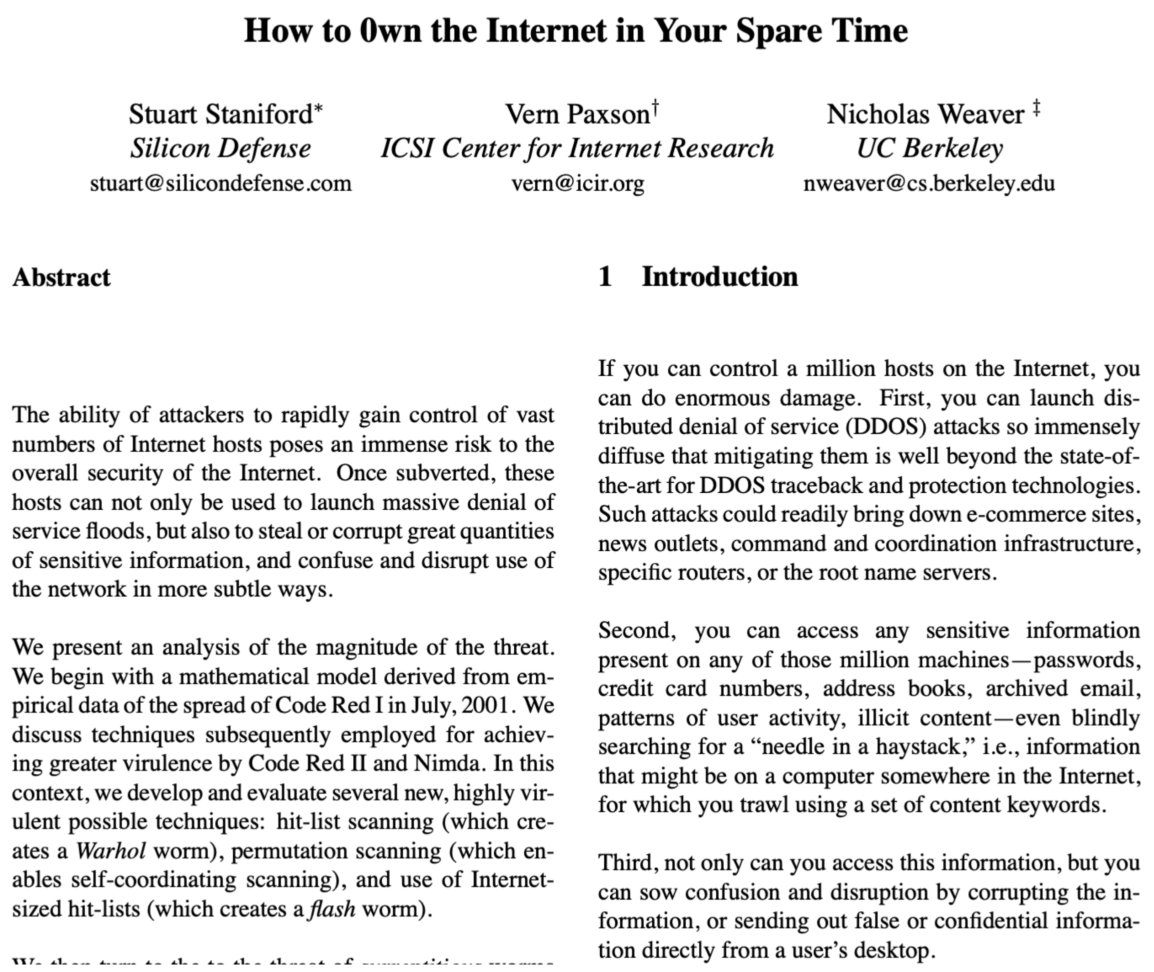
With many thanks to Ross Anderson and Alastair R. Beresford

Okay Google, what’s a Whopper?

The Morris Worm: breaking into computers at scale (1988)

- Exploited vulnerabilities in sendmail, fingerd, rsh
- Used a list of common weak passwords
- Gov. assessment: $100k to $10M in damage
- 6,000* machines infected
- Internet partitioned for days to prevent reinfection
- Robert Morris was the first person convicted under the 1986 Computer Fraud and Misuse Act.
  - 3 year suspended sentence
  - 400 hr community service
  - $10k fine.

"How to 0wn the Internet in your spare time" - Usenix Security 2002



How to 0wn the Internet in Your Spare Time

Stuart Staniford[*]
Silicon Defense
stuart@silicondefense.com

Vern Paxson[†]
ICSI Center for Internet Research
vern@icir.org

Nicholas Weaver [‡]
UC Berkeley
nweaver@cs.berkeley.edu

Abstract

The ability of attackers to rapidly gain control of vast numbers of Internet hosts poses an immense risk to the overall security of the Internet. Once subverted, these hosts can not only be used to launch massive denial of service floods, but also to steal or corrupt great quantities of sensitive information, and confuse and disrupt use of the network in more subtle ways.

We present an analysis of the magnitude of the threat. We begin with a mathematical model derived from empirical data of the spread of Code Red I in July, 2001. We discuss techniques subsequently employed for achieving greater virulence by Code Red II and Nimda. In this context, we develop and evaluate several new, highly virulent possible techniques: hit-list scanning (which creates a *Warhol* worm), permutation scanning (which enables self-coordinating scanning), and use of Internet-sized hit-lists (which creates a *flash* worm).

We then turn to the to the threat of surreptitious worms

1 Introduction

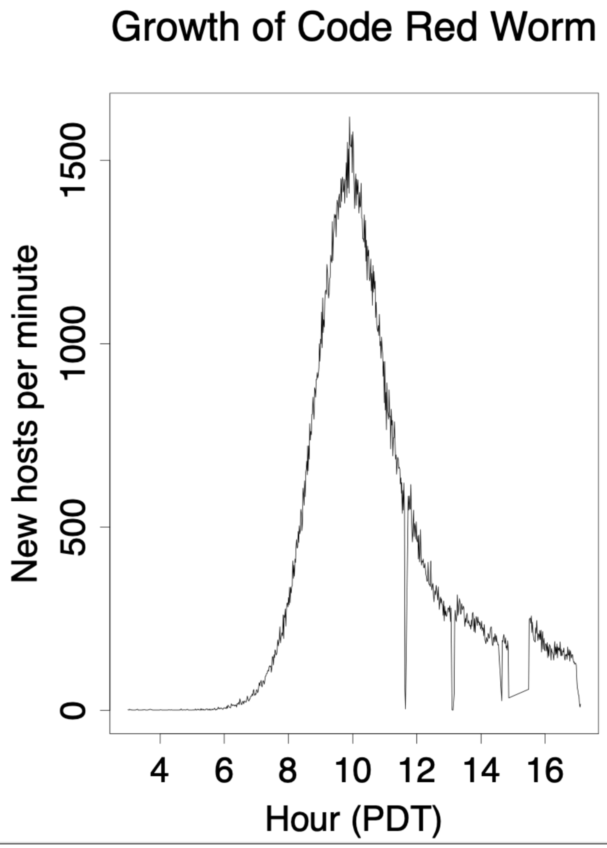
If you can control a million hosts on the Internet, you can do enormous damage. First, you can launch distributed denial of service (DDOS) attacks so immensely diffuse that mitigating them is well beyond the state-of-the-art for DDOS traceback and protection technologies. Such attacks could readily bring down e-commerce sites, news outlets, command and coordination infrastructure, specific routers, or the root name servers.

Second, you can access any sensitive information present on any of those million machines—passwords, credit card numbers, address books, archived email, patterns of user activity, illicit content—even blindly searching for a "needle in a haystack," i.e., information that might be on a computer somewhere in the Internet, for which you trawl using a set of content keywords.

Third, not only can you access this information, but you can sow confusion and disruption by corrupting the information, or sending out false or confidential information directly from a user's desktop.

https://www.usenix.org/legacy/event/sec02/full_papers/staniford/staniford.pdf

"Code Red" worm. July 2001.



Spread of Code Red:

Monitoring two class B's $\Rightarrow$ 300,000 infected hosts.

Analytic model:

$N =$ total number of vulnerable hosts

$K =$ compromise rate, new hosts/host/hour

$a(t) =$ proportion of vulnerable machines compromised at time t

Then:

$$a(t) = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}}$$

$\Rightarrow$ *logistic* growth.

Source: https://www.icir.org/vern/talks/vp-0wn-UCB.pdf

"Slammer" worm. 2003.

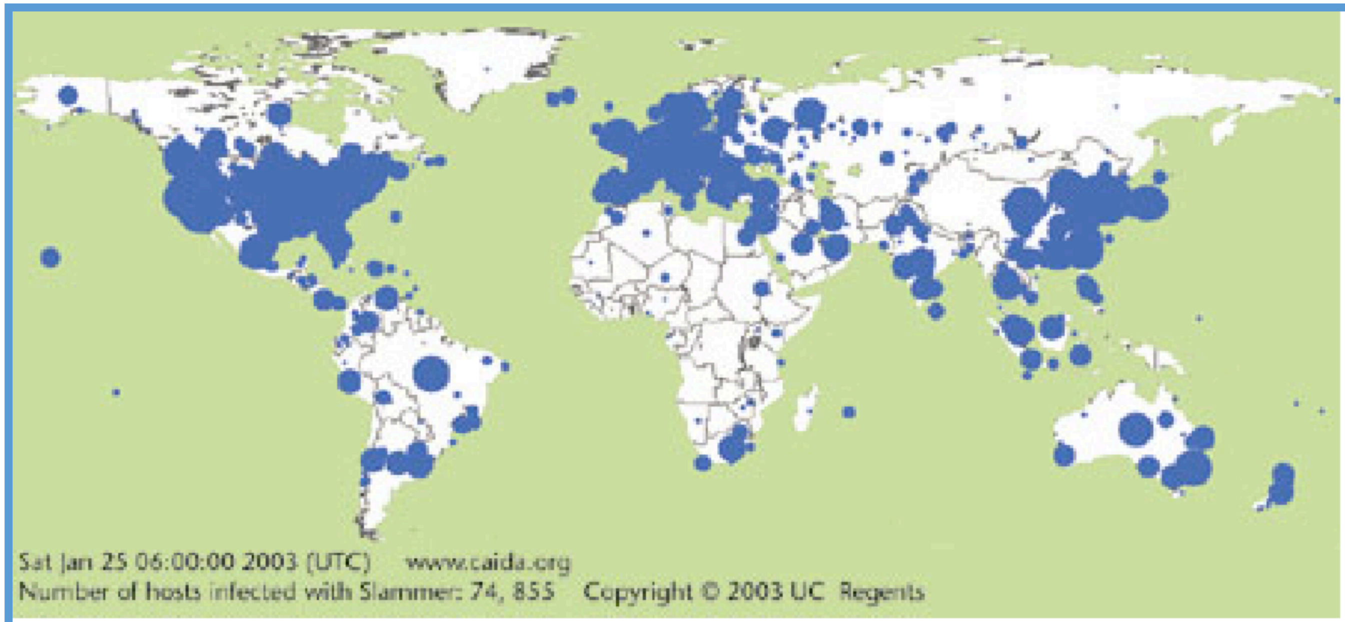
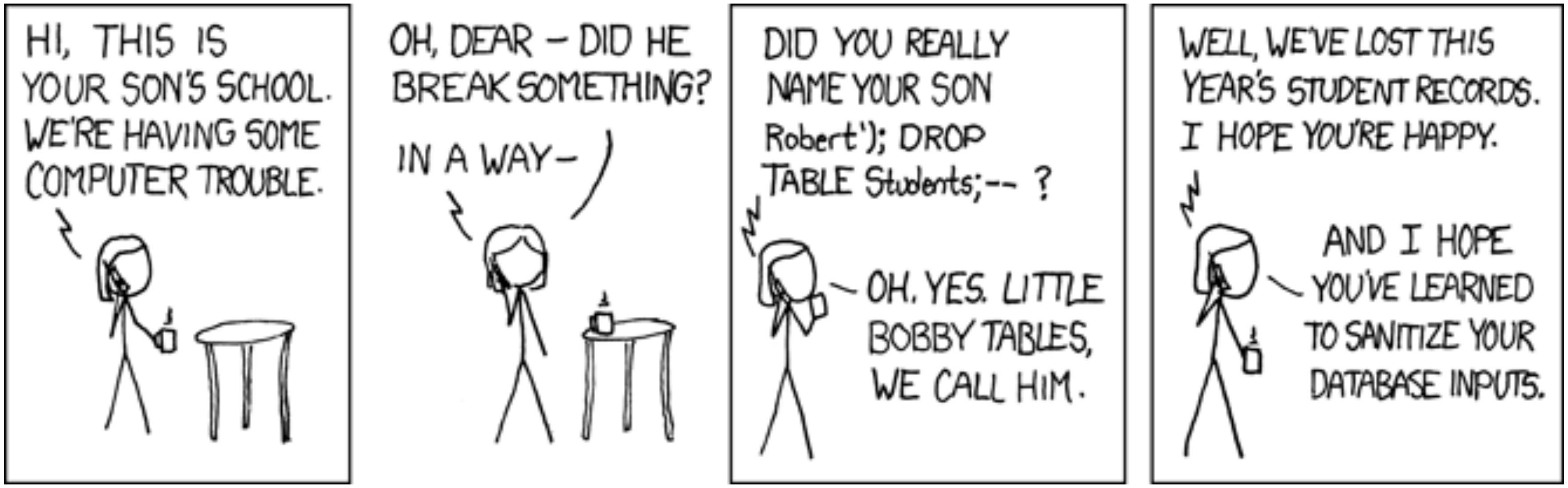

Sat Jan 25 06:00:00 2003 (UTC) www.caida.org
Number of hosts infected with Slammer: 74, 855 Copyright © 2003 UC Regents

Figure 1. The geographical spread of Slammer in the 30 minutes after its release. The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles

SQL Injection attack: failure to sanitize untrusted inputs


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

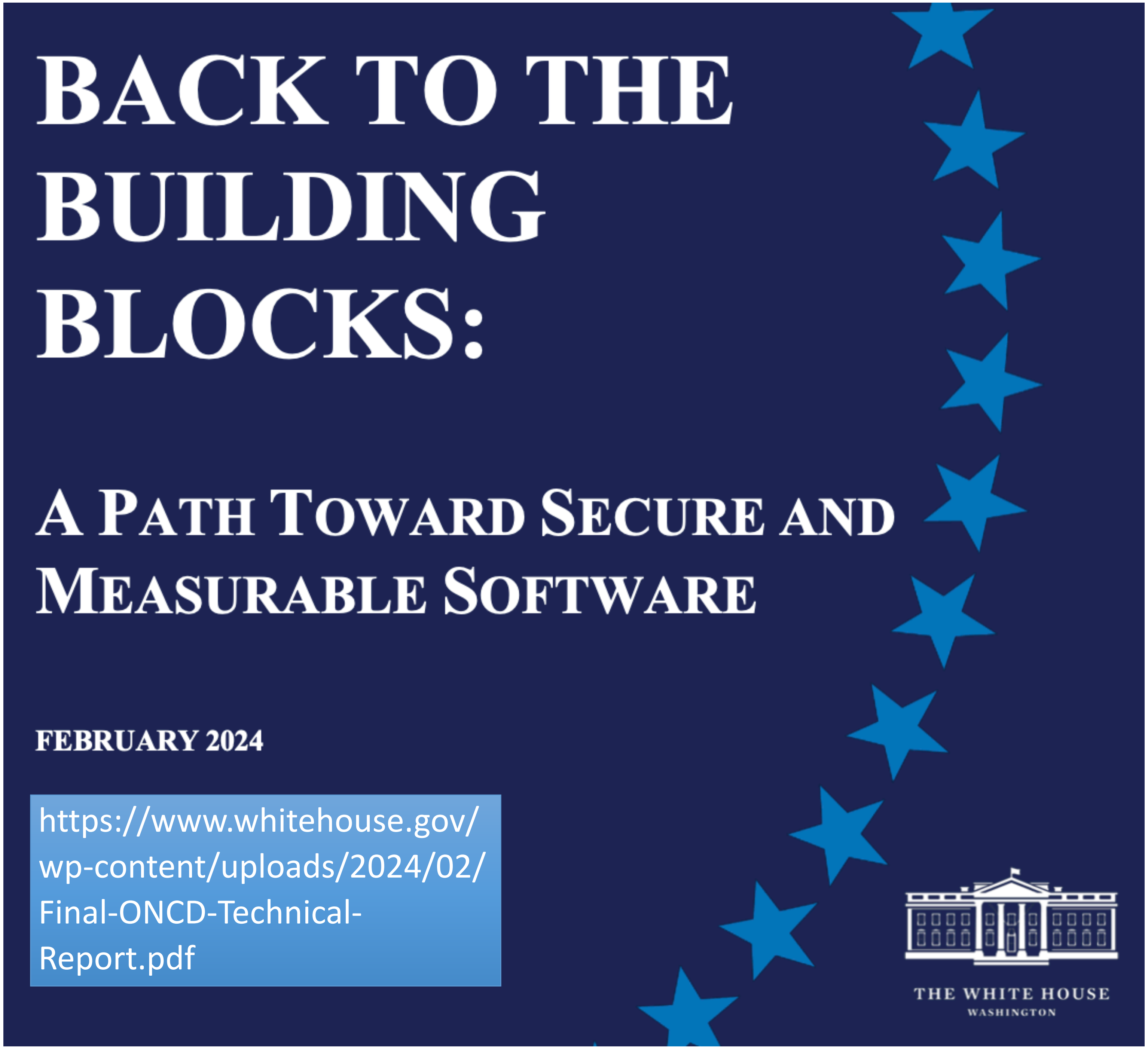
```
String sql =
  "INSERT INTO Students (Name) VALUES ('"
  + studentName
  + "');";
```

Software countermeasures: systems and tools

- Operating system protections
  - Data execution prevention
  - Address space layout randomisation
  - …
- Tools, e.g. Coverity
  - Static analysis
  - Dynamic analysis
  - Testing frameworks
  - …
- Automated update systems to install patches

Software countermeasures: reducing bug number and severity

- Defensive programming
- Secure coding standards
  - See Howard and LeBlanc on MS standards for C
- Contracts, e.g. in the Eiffel language
- API analysis
  - Combining API calls may lead to vulnerabilities
  - Challenging for APIs accessible over the Internet



BACK TO THE BUILDING BLOCKS:

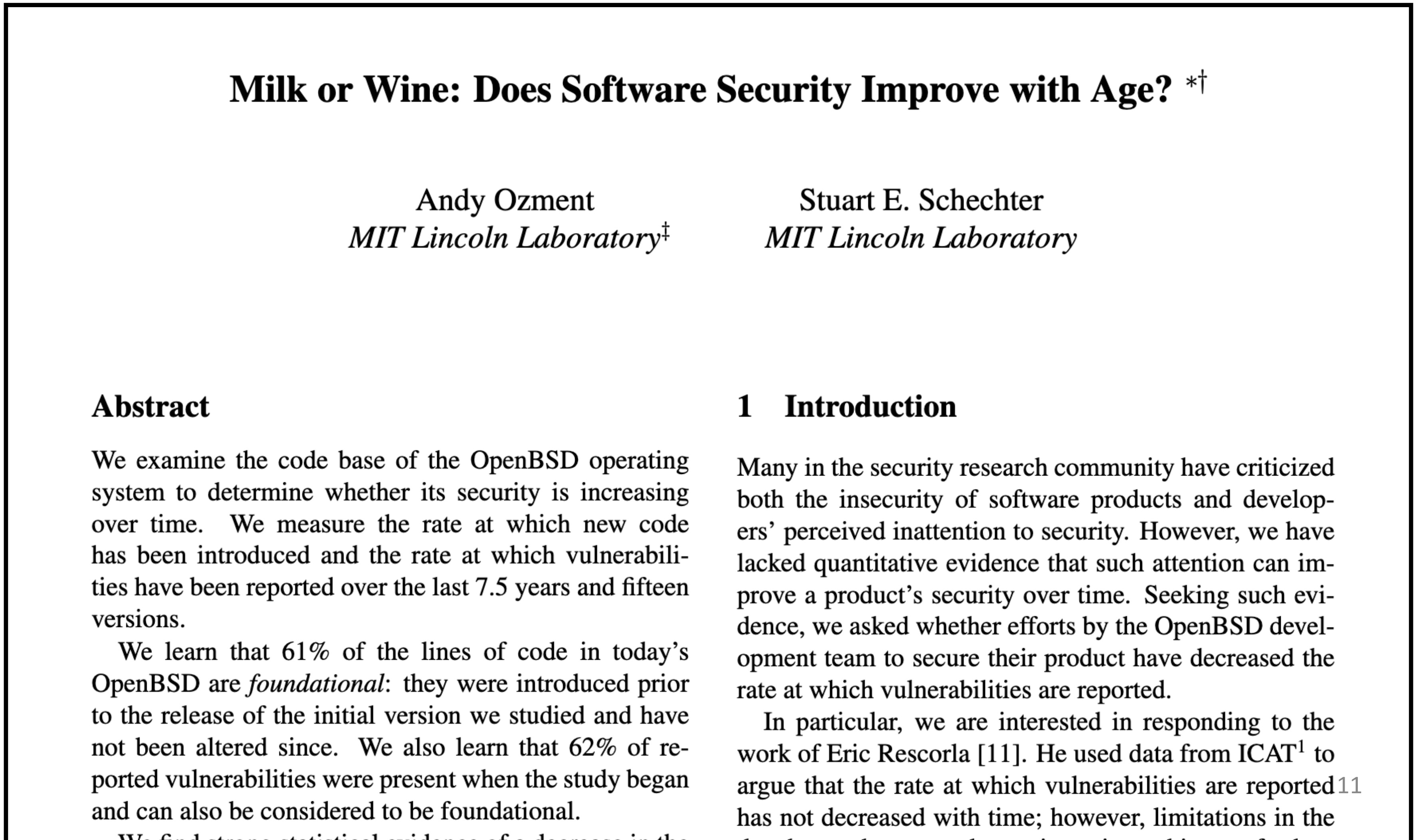
A PATH TOWARD SECURE AND MEASURABLE SOFTWARE

FEBRUARY 2024

THE WHITE HOUSE
WASHINGTON

https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf

We cannot write code without latent vulnerabilities



Milk or Wine: Does Software Security Improve with Age? *†

Andy Ozment
MIT Lincoln Laboratory‡

Stuart E. Schechter
MIT Lincoln Laboratory

Abstract

We examine the code base of the OpenBSD operating system to determine whether its security is increasing over time. We measure the rate at which new code has been introduced and the rate at which vulnerabilities have been reported over the last 7.5 years and fifteen versions.

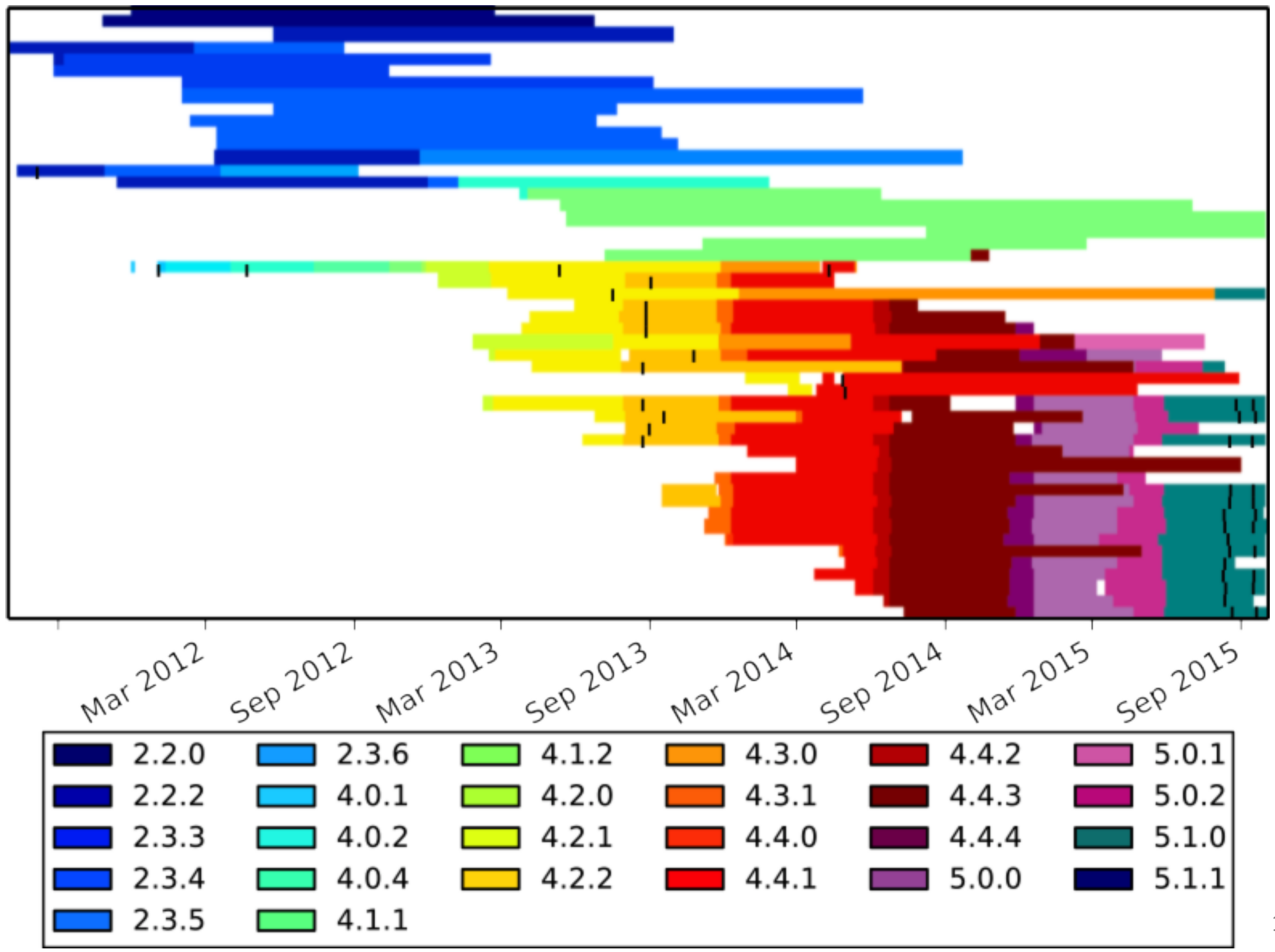
We learn that 61% of the lines of code in today's OpenBSD are *foundational*: they were introduced prior to the release of the initial version we studied and have not been altered since. We also learn that 62% of reported vulnerabilities were present when the study began and can also be considered to be foundational.

1 Introduction

Many in the security research community have criticized both the insecurity of software products and developers' perceived inattention to security. However, we have lacked quantitative evidence that such attention can improve a product's security over time. Seeking such evidence, we asked whether efforts by the OpenBSD development team to secure their product have decreased the rate at which vulnerabilities are reported.

In particular, we are interested in responding to the work of Eric Rescorla [11]. He used data from ICAT[1] to argue that the rate at which vulnerabilities are reported has not decreased with time; however, limitations in the

OS versions of 50 LG handsets

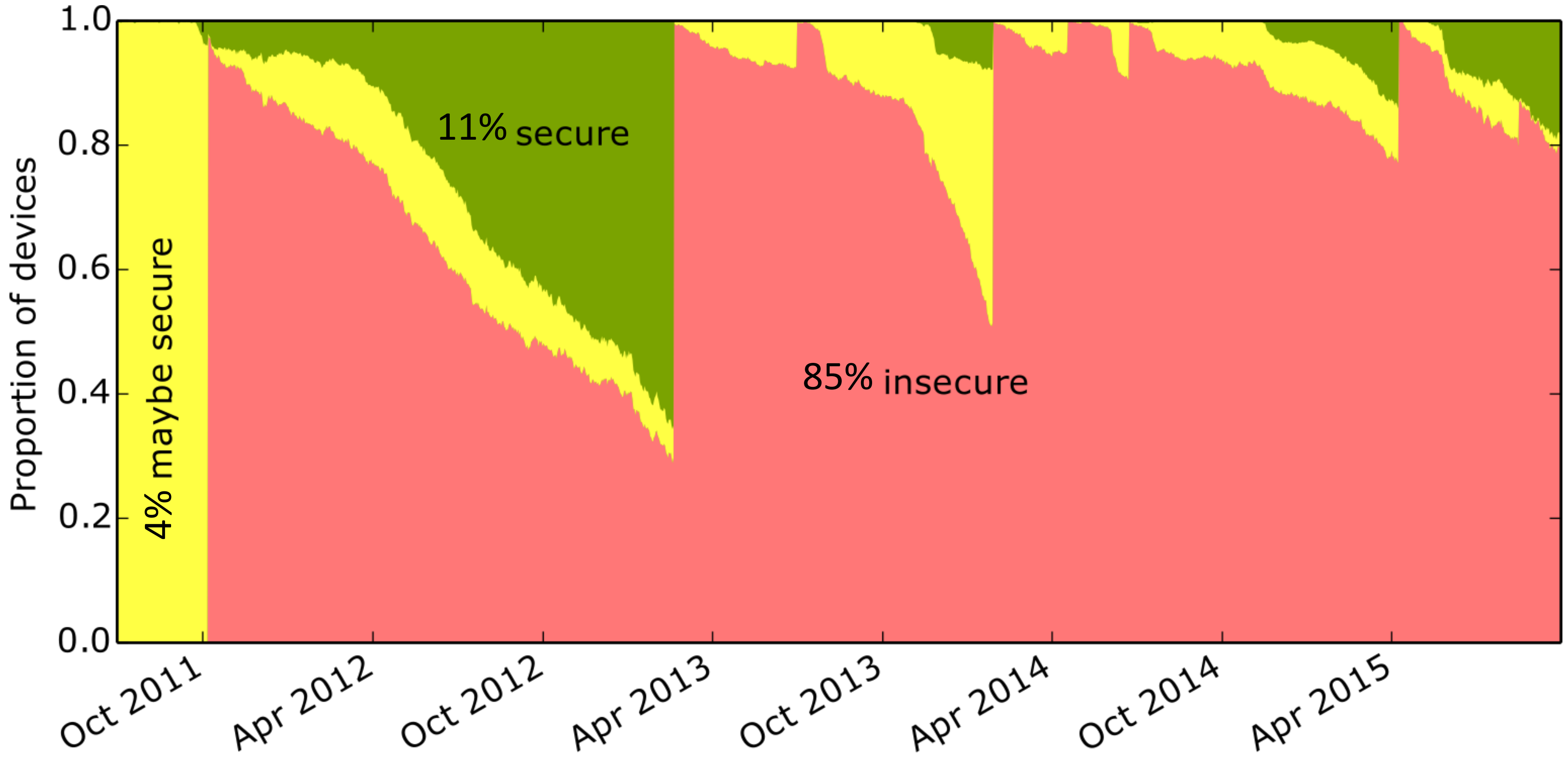


Link OS versions to database of vulnerabilities

Match OS version information to OS and Build Number to put each handset into one group:

- Insecure
- Maybe secure
- Secure

On average, 85% are vulnerable



The *Software Crisis*

- Software still lags behind hardware’s potential
- Many large projects are late, over budget, dysfunctional, or abandoned (CAPSA, NPfIT, DWP, Addenbrookes, …)
- Some failures cost lives (Therac 25) or billions (Ariane 5, NPfIT)
- Some expensive scares (Y2K, Pentium)
- Some combine the above (LAS)

London Ambulance Service disaster

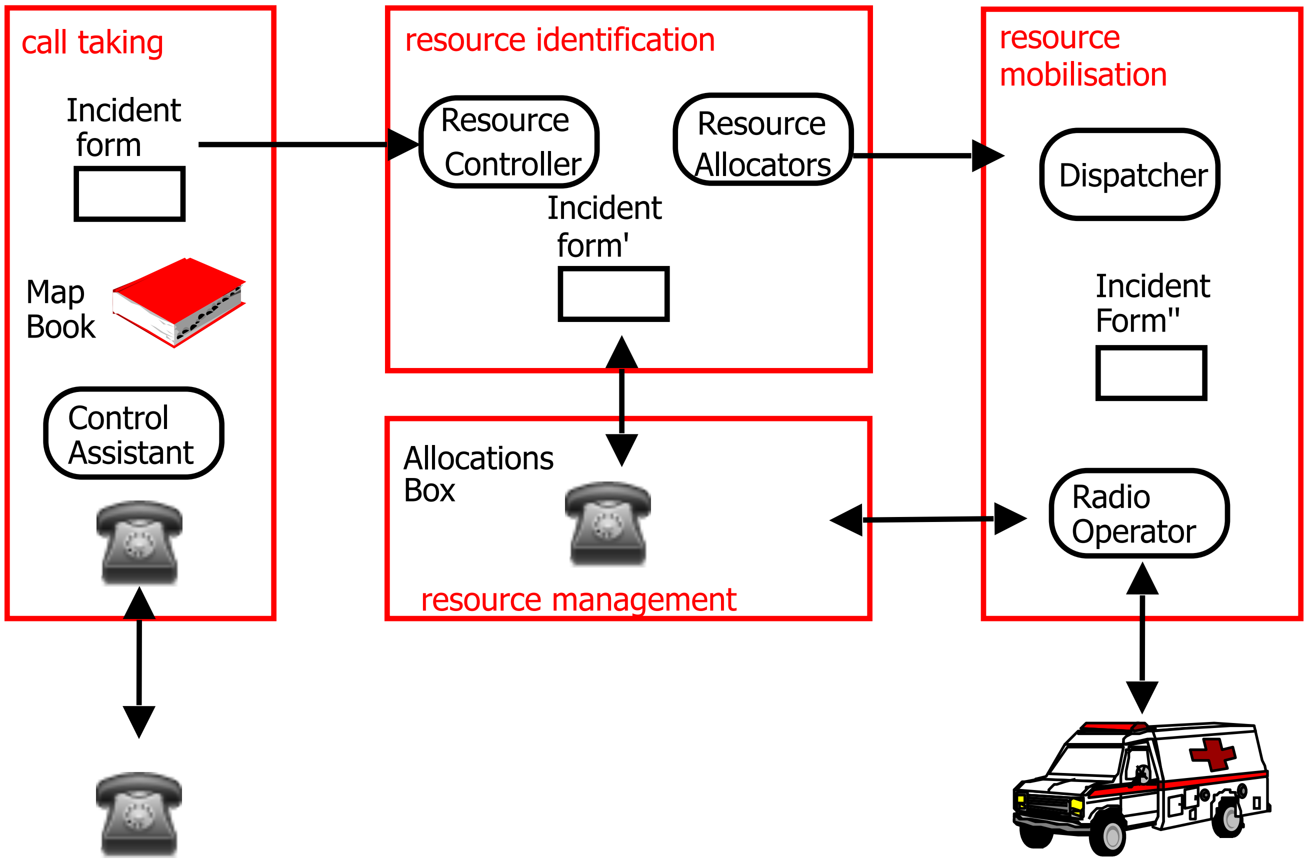
- Widely cited example of project
- Many aspects of the failure widely repeated since

- Attempt to automate ambulance dispatch in 1992
- Result left London without service for a day
- Number estimated deaths ran as high as 20
- CEO sacked; public outrage

Project background

- Attempt to automate in 1980s failed – system failed load test
- Industrial relations poor; pressure to cut costs
- Public concern over service quality
- South West Thames Regional Health Authority decided on fully automated system: responder would “email” ambulance
- Consultancy study said this might cost £1.9m and take 19 months, *provided a packaged solution could be found*. AVLS would be extra

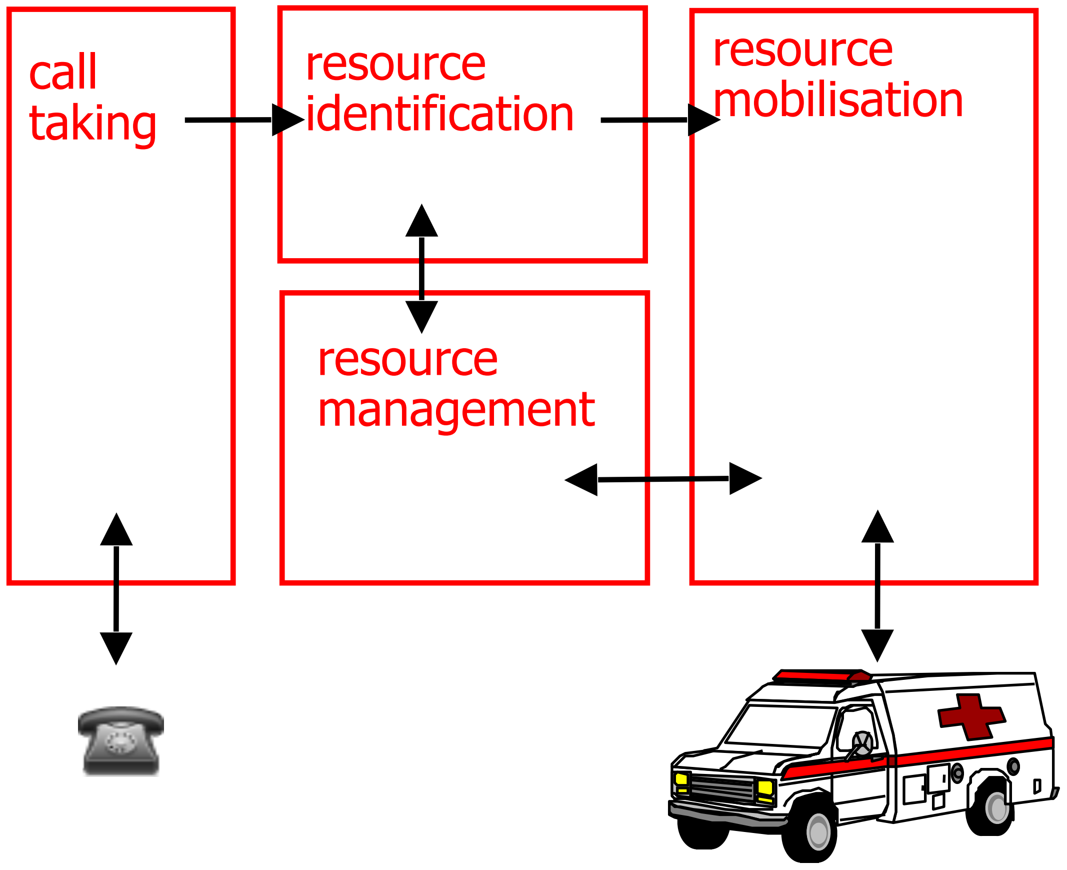
Original dispatch system worked on paper with regional control



Many problems with original system

- It took 3 minutes to dispatch an ambulance
- It required 200 staff (out of 2700 in total).
- There were errors, especially in deduplication
- Queues and bottlenecks, especially with the radio
- Call-backs tiresome

Computer-aided dispatch system



- Large
- Real-time
- Critical
- Data rich
- Embedded
- Distributed
- Mobile components

Tender process was poor

- Idea of a £1.5m system stuck; idea of AVLS added; proviso of a packaged solution forgotten; new IS director hired
- Tendered on 7th Feb 1991; completion due Jan 1992
- 35 firms looked at tender; 19 proposed; most said timescale unrealistic, only partial automation possible by early 1992
- Tender awarded to consortium of Systems Options Ltd, Apricot and Datatrak for £937,463
    - £700K cheaper than next lowest bidder!

Phase one: design work ‘done’ in July and contract signed in August

Minutes of a progress meeting in June recorded:

- A 6-month timescale for an 18-month project
- A lack of methodology
- No full-time LAS users providing domain knowledge
- Lead contractor (System Options) relied heavily on cozy assurances of subcontractors

Unsurprisingly LAS told in December that only partial automation by January deadline – front end for call taking, gazetteer, docket printing

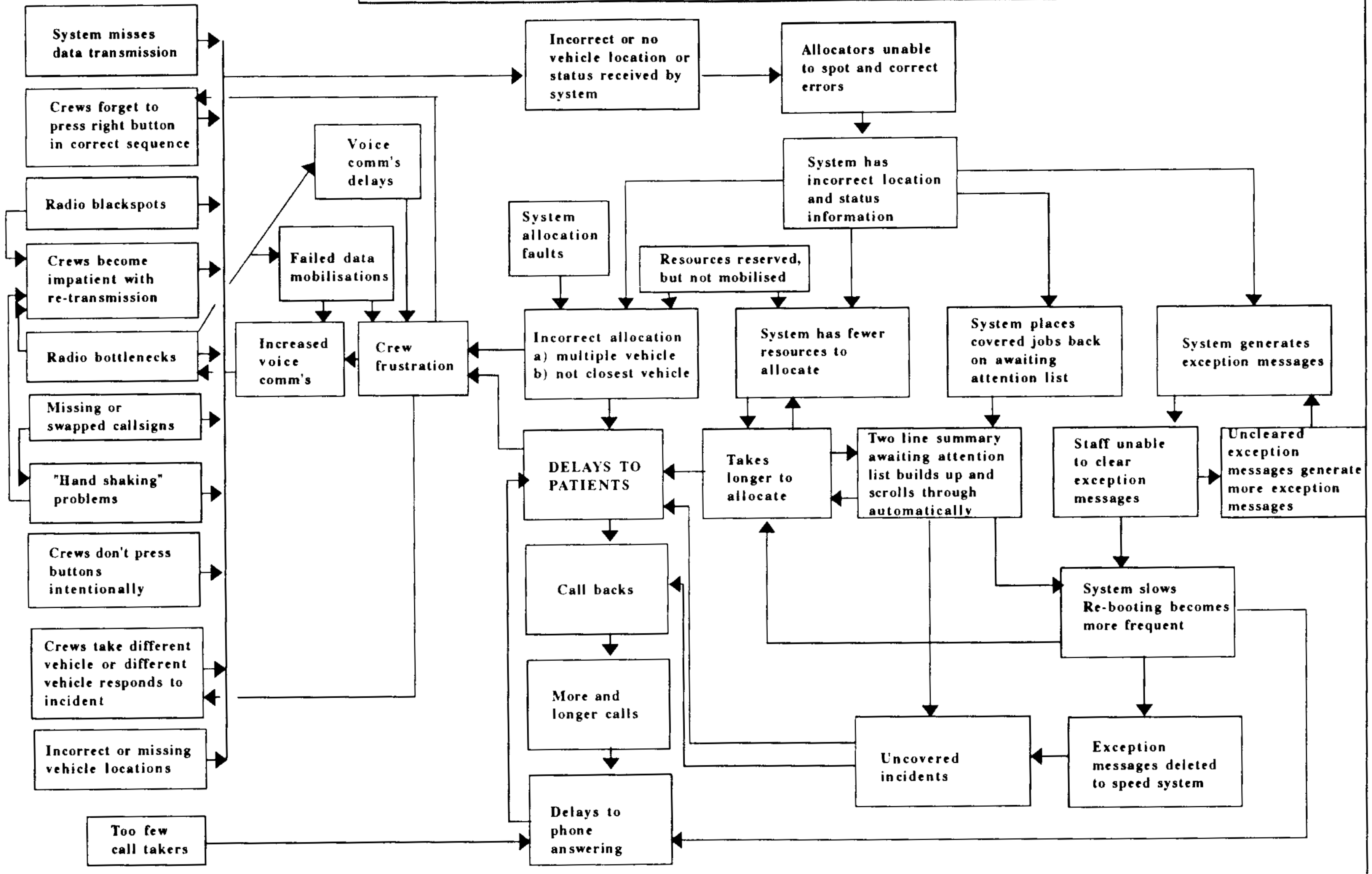
Phase two: full automation

- Server never stable in 1992; client and server lockup
- Radio messaging with blackspots and congestion; couldn’t cope with established working practices
- Management decided to go live on 26th Oct 1992
- Independent review had called for volume testing, implementation strategy, change control, …all ignored
- CEO: “No evidence to suggest that the full system software, when commissioned, will not prove reliable”
- On 26 Oct 1992, room was reconfigured to use terminals, not paper. There was no backup…

Circle of disaster on 26/7th October

- System progressively lost track of vehicles
- Exception messages scrolled off screen and were lost
- Incidents held as allocators searched for vehicles
- Callbacks from patients increased causing congestion
- data delays → voice congestion → crew frustration → pressing wrong buttons and taking wrong vehicles → many vehicles sent to an incident, or none
- System slowdown and congestion leading to collapse

Diagram 4.5
26/27 October Cause/Effect Diagram


System misses data transmission
Crews forget to press right button in correct sequence
Radio blackspots
Crews become impatient with re-transmission
Radio bottlenecks
Missing or swapped callsigns
"Hand shaking" problems
Crews don't press buttons intentionally
Crews take different vehicle or different vehicle responds to incident
Incorrect or missing vehicle locations
Too few call takers
Voice comm's delays
Failed data mobilisations
Increased voice comm's
Crew frustration
Incorrect or no vehicle location or status received by system
Allocators unable to spot and correct errors
System has incorrect location and status information
System allocation faults
Resources reserved, but not mobilised
Incorrect allocation a) multiple vehicle b) not closest vehicle
System has fewer resources to allocate
System places covered jobs back on awaiting attention list
System generates exception messages
DELAYS TO PATIENTS
Takes longer to allocate
Two line summary awaiting attention list builds up and scrolls through automatically
Staff unable to clear exception messages
Uncleared exception messages generate more exception messages
Call backs
System slows Re-booting becomes more frequent
More and longer calls
Uncovered incidents
Exception messages deleted to speed system
Delays to phone answering

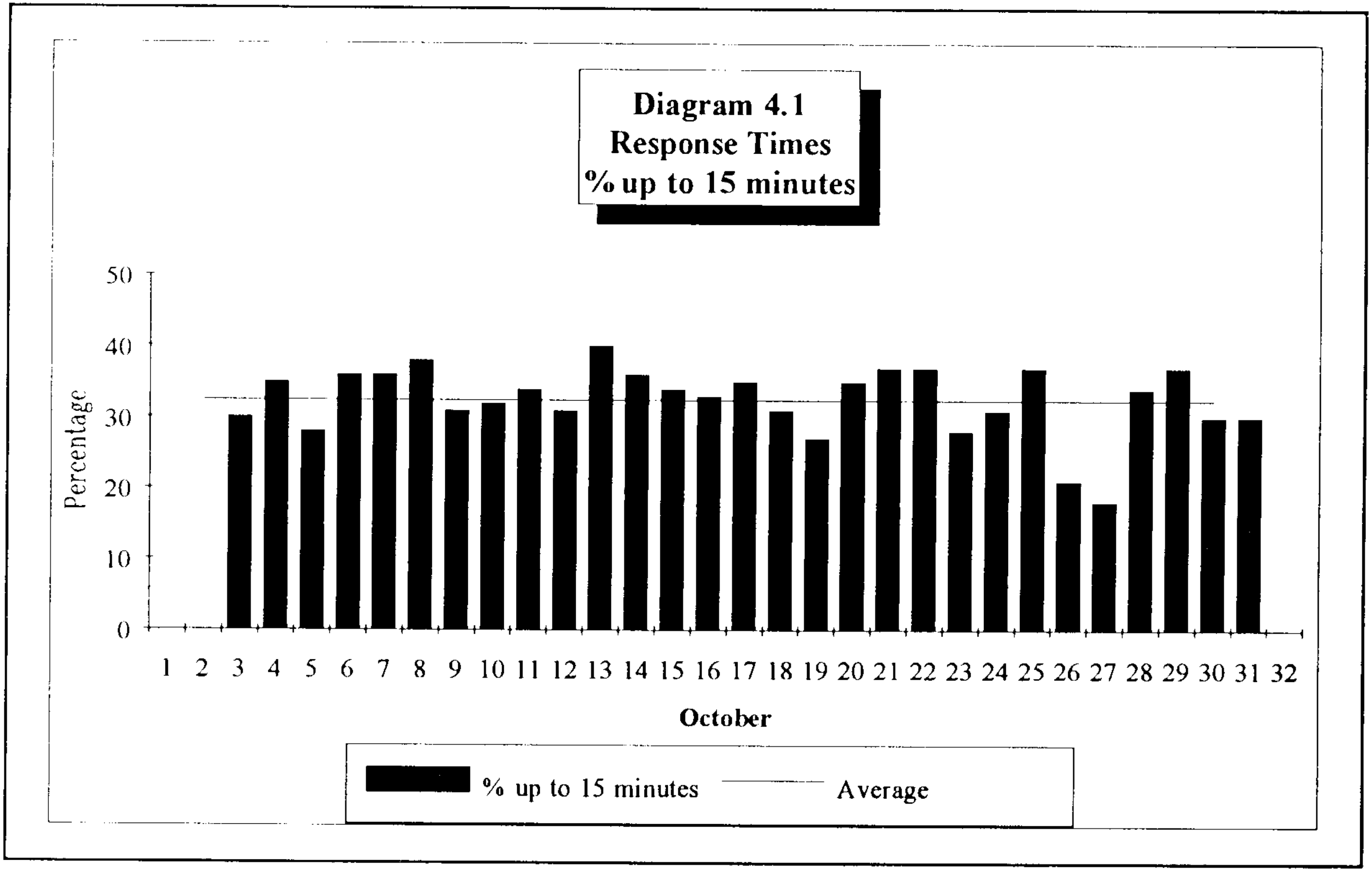


Diagram 4.1
Response Times
% up to 15 minutes
50
40
30
20
10
0
Percentage
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
October
% up to 15 minutes
Average

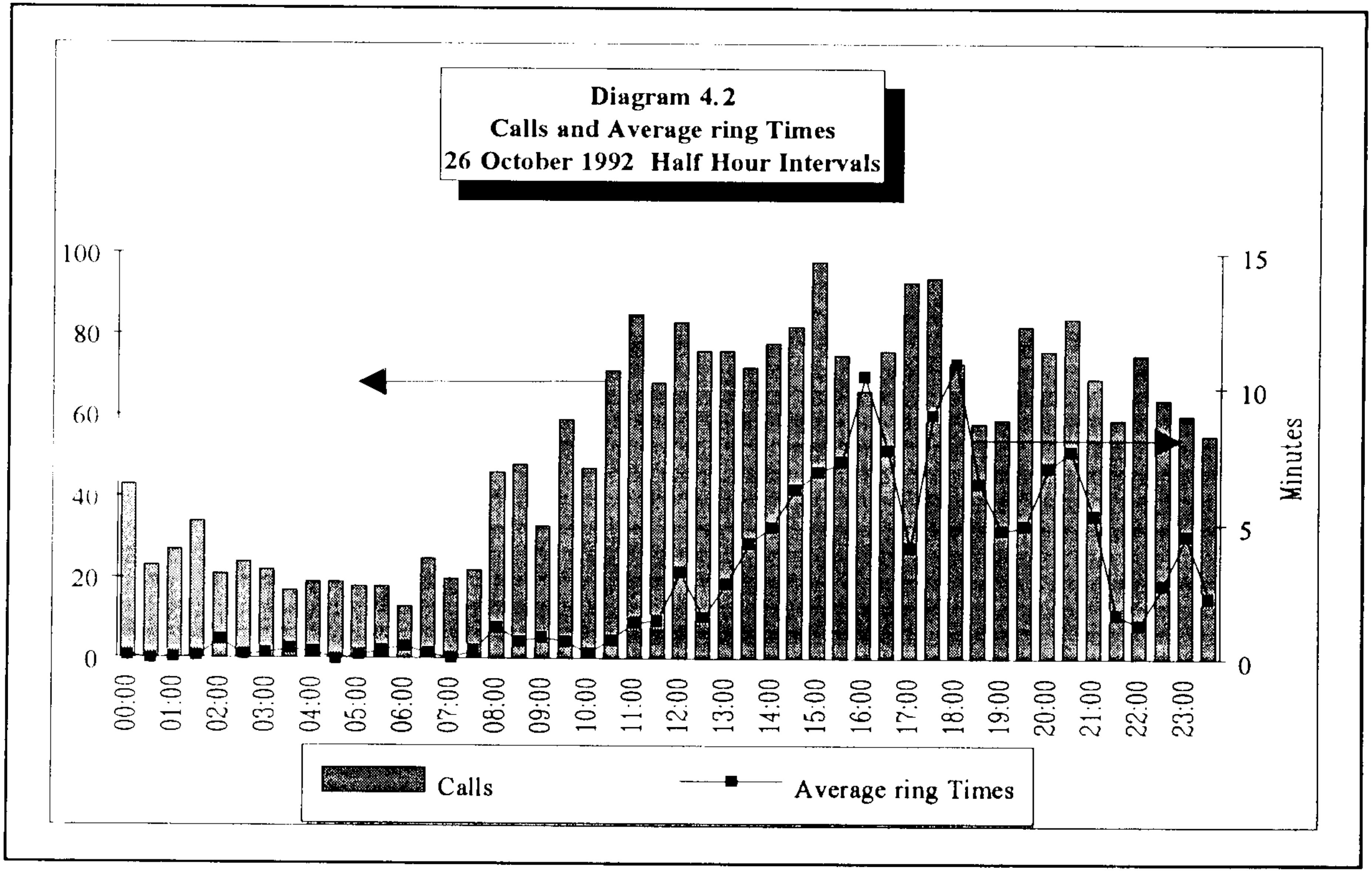


Diagram 4.2
Calls and Average ring Times
26 October 1992 Half Hour Intervals
100
80
60
40
20
0
15
10
5
0
Minutes
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Calls
Average ring Times

Diagram 4.3
Total A&E Patients Carried
October

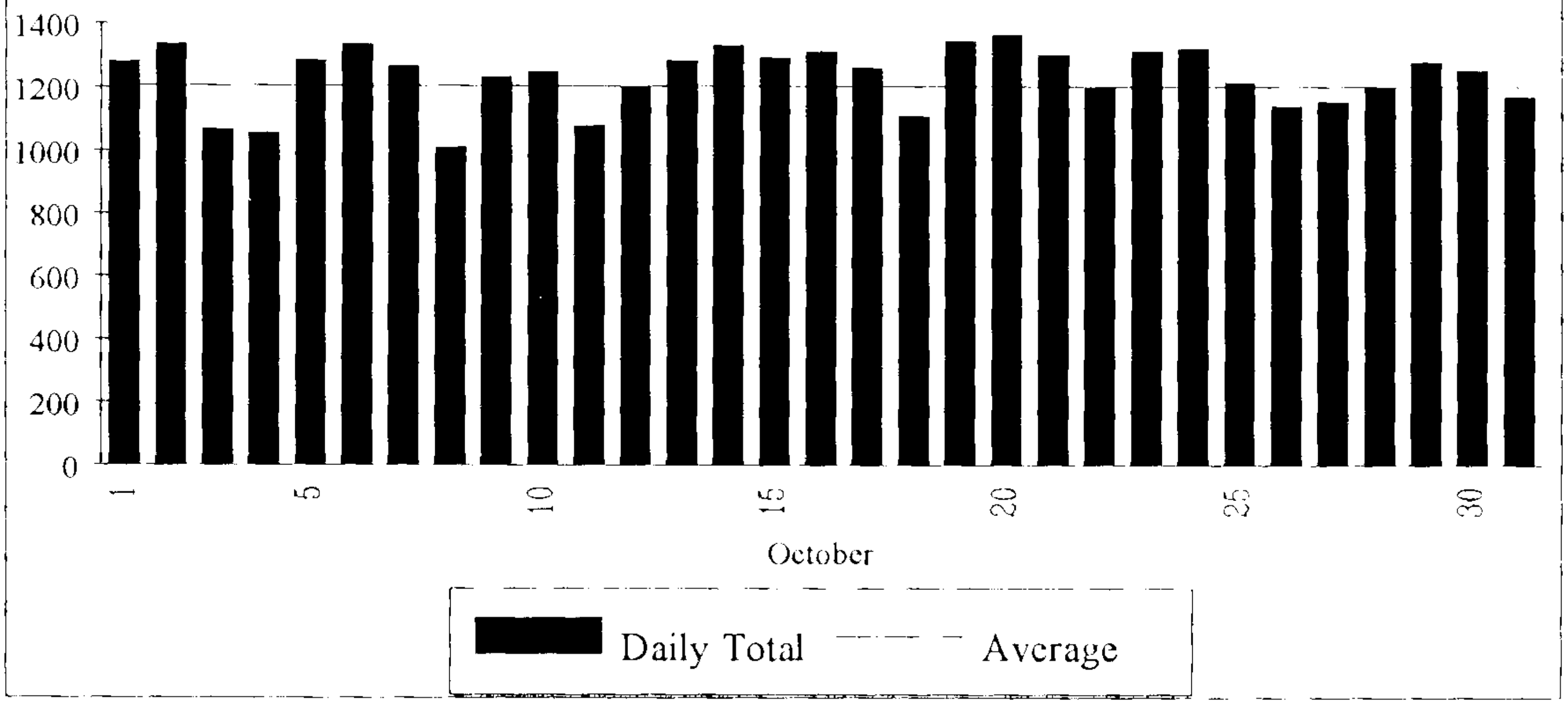
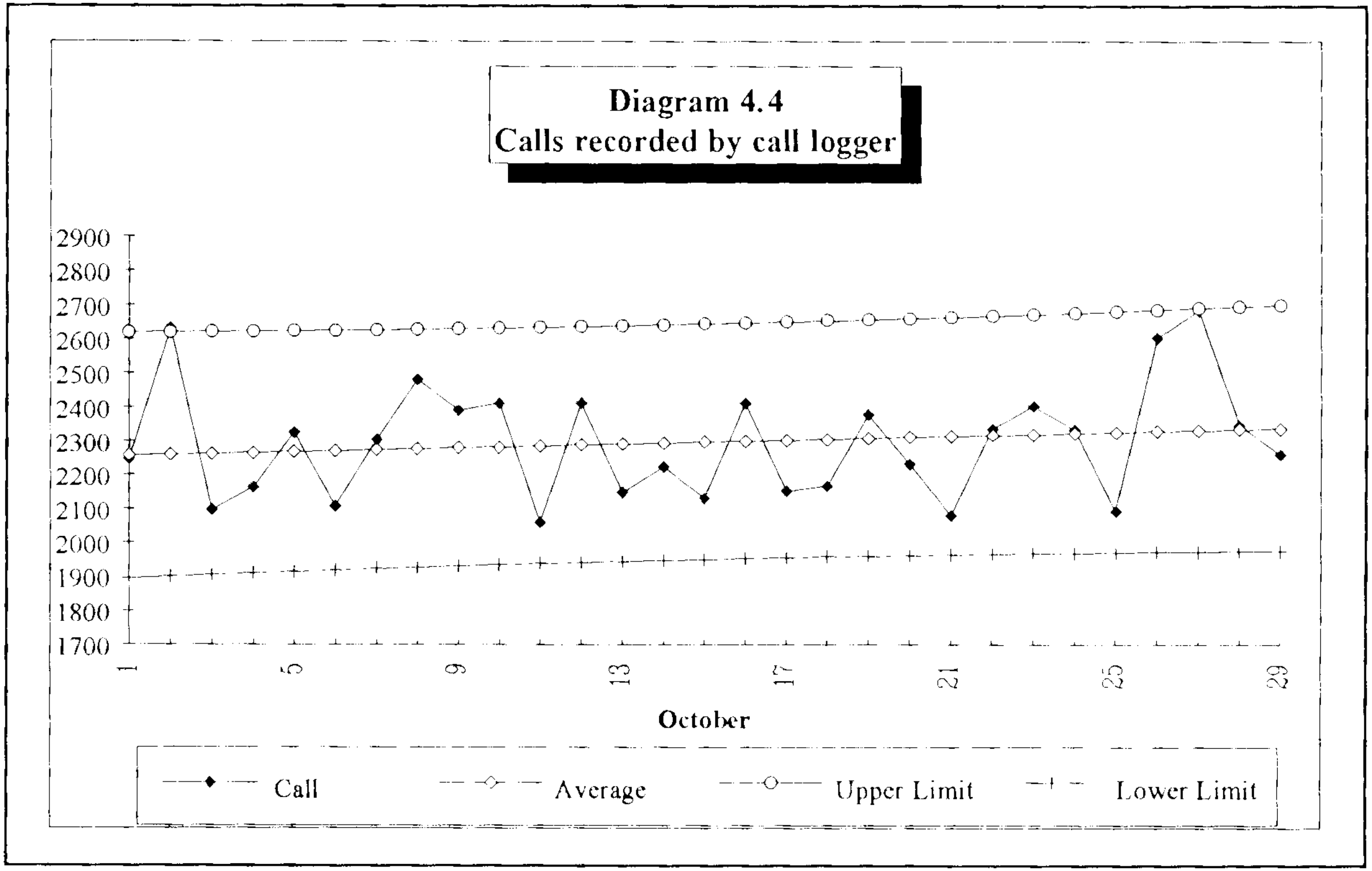


Diagram 4.4
Calls recorded by call logger



Collapse likely resulted in deaths

- One ambulance arrived to find the patient dead and taken away by undertakers
- Another answered a ‘stroke’ call after 11 hours and 5 hours after the patient had made their own way to hospital
- …
- Chief executive resigns