"Parasitic Computing"

Seminar by:

Kunal Goswami

05IT6006

Outline

- Introduction
- How it differs from others
- Internet Communication
- Proof of Concept
- 2-SAT problem
- Implementation using TCP
- Issues
  - Problems for parasites
  - Problems for servers
- References

Introduction

- First reported in Journal “Nature” in 2001 by Barabasi, Freeh, Jeong and Brockman
- A form of distributed computing
- Used for solving complex computational problems
- Exploits standard set of communication protocol on internet

How it differs from other?

- Cluster Computing
  - In cluster computing, many computers pool in their resources willingly
  - Parasitic computing does not require the willingness of any target machine
- It is not cracking

Internet Communication

When user selects a URL

- Actions at sender
  - Open a TCP connection to a web server
  - Issue a HTTP request over TCP connection
  - TCP message is carried via IP
- Actions at receiver
  - Receive message through IP
  - Validate checksum at TCP
    - Validated pushed to HTTP
    - Not validated discard the packet

Internet Communication

- HTTP → TCP → IP → TCP → HTTP

Proof of concept

- Solved the SAT problem
- NP-Complete
- Usually solved by testing several candidate solutions
- Each candidate solution can be examined in parallel

2-SAT problem

$$\mathbf{P=}\ (X_1 \oplus X_2) \wedge (X_3 \wedge X_4) \wedge (X_5 \oplus X_6) \wedge (X_7 \oplus X_8)$$
$$\wedge$$
$$(X_9 \wedge X_{10}) \wedge (X_{11} \oplus X_{12}) \wedge (X_{13} \wedge X_{14}) \wedge (X_{15} \oplus X_{16})$$

X	Y	$X \oplus Y$	$X \wedge Y$	$X+Y$
0	0	0	0	0 0
0	1	1	0	0 1
1	0	1	0	0 1
1	1	0	1	1 0

Implementation

- Computation takes place at many layers in Internet
- Several Internet protocols could be exploited to perform Parasitic computation
- Implementation made using TCP or higher levels

Implementation using TCP

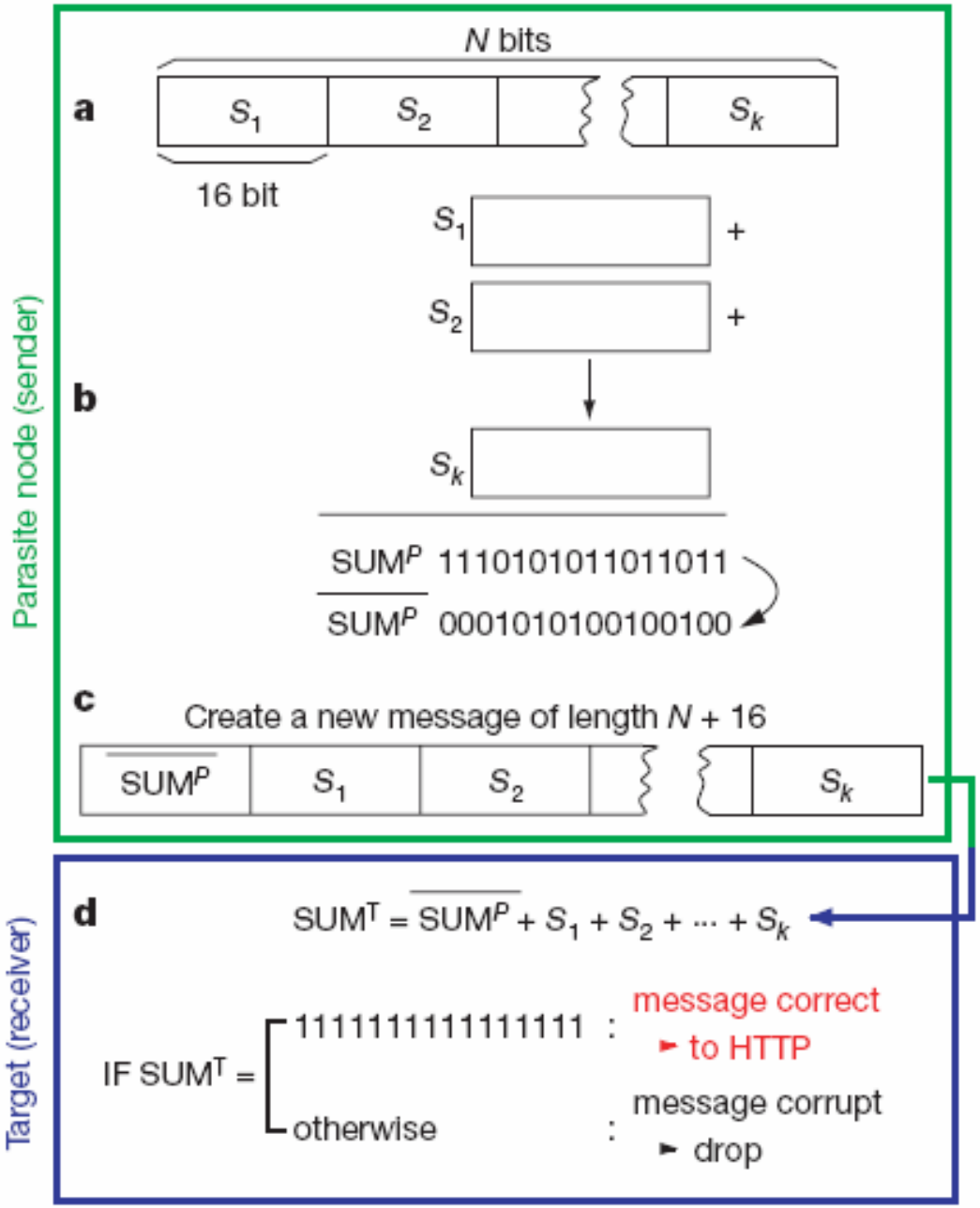
- TCP Checksums
  - Provides enough logic to compute any boolean function.
- HTTP service
  - Protocol always send a response to any message received

Solving 2-SAT problem

- Send a specially created TCP packet containing a possible solution.
- If the possible solution is correct, the HTTP server returns an error message.
- If the solution is wrong, then packet is dropped at TCP layer.

TCP checksum

- Sender breaks message into 16 bit words
- These words are added together
- The result is inverted and sent with message
- Receiver breaks message into 16 bit words
- All words are added together
- If the result is all ones, the message is valid
- Otherwise, the message is dropped


Parasite node (sender)
a
N bits
S_1
S_2
S_k
16 bit
b
S_1 +
S_2 +
S_k
SUM^P 1110101011011011
$\overline{SUM^P}$ 0001010100100100
c
Create a new message of length N + 16
$\overline{SUM^P}$
S_1
S_2
S_k
Target (receiver)
d
$SUM^T = \overline{SUM^P} + S_1 + S_2 + \cdots + S_k$
IF SUM^T =
1111111111111111 : message correct
► to HTTP
otherwise : message corrupt
► drop

Creating parasitic messages

- Variables values of a possible solution are distributed so they will align when split in 16 byte words
- TCP checksum sent is a representation of the expected answer
- “Parasitic” checksum and the variable string are sent to remote machine for verification

a

$$P = (x_1 \oplus x_2) \wedge (x_3 \wedge x_4) \wedge (x_5 \oplus x_6) \wedge (x_7 \oplus x_8) \wedge (x_9 \wedge x_{10}) \wedge (x_{11} \oplus x_{12}) \wedge (x_{13} \wedge x_{14}) \wedge (x_{15} \oplus x_{16})$$

b

X	Y	$X \oplus Y$	$X \wedge Y$	$X + Y$
0	0	0	0	0 0
0	1	1	0	0 1
1	0	1	0	0 1
1	1	0	1	1 0

c

$M = 0x_1\ 0x_3\ 0x_5\ 0x_7\ 0x_9\ 0x_{11}\ 0x_{13}\ 0x_{15}$ | $0x_2\ 0x_4\ 0x_6\ 0x_8\ 0x_{10}\ 0x_{12}\ 0x_{14}\ 0x_{16}$

$E = 01\ 00\ 01\ 01\ 00\ 01\ 01\ 01$ | $00\ 00\ 01\ 00\ 01\ 01\ 01\ 00$

S_1 | S_2

d

$0x_1\ 0x_3\ 0x_5\ 0x_7\ 0x_9\ 0x_{11}\ 0x_{13}\ 0x_{15}$

$0x_2\ 0x_4\ 0x_6\ 0x_8\ 0x_{10}\ 0x_{12}\ 0x_{14}\ 0x_{16}$

$\oplus\ \wedge\ \oplus\ \oplus\ \wedge\ \oplus\ \wedge\ \oplus$

01 10 01 01 10 01 10 01

e

S_1	01 00 01 01 00 01 01 01
S_2	00 00 01 00 01 01 01 00
SUM	01 00 10 01 01 10 10 01
$\overline{\text{SUM}}$	10 11 01 10 10 01 01 10

(Real checksum)

10 01 10 10 01 10 01 10

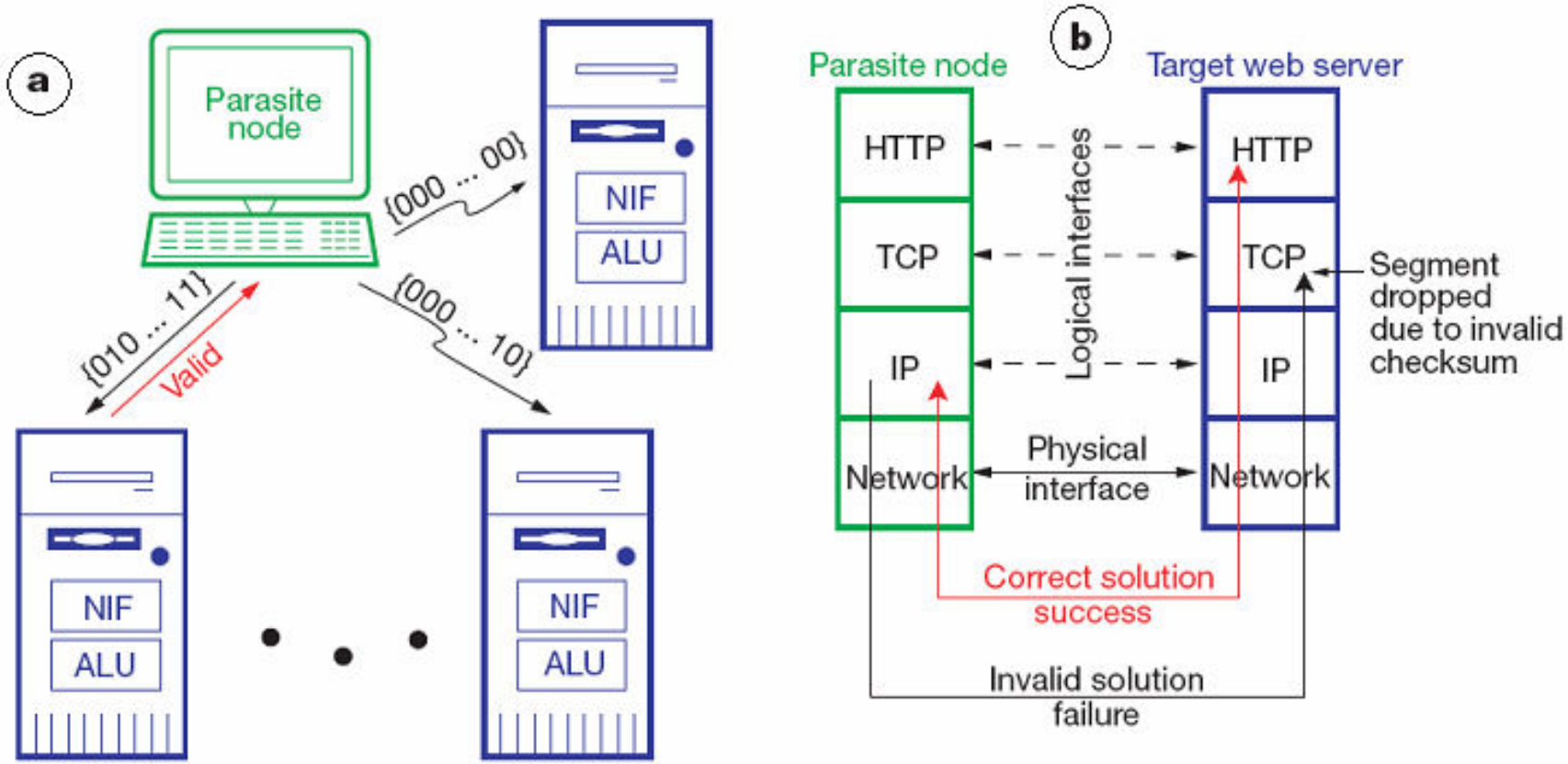
T_c

f

Transmitted message

1001101001100110	0100010100010101	0000010001010100
T_c	S_1	S_2

Prototype of parasitic computer



3-SAT Problem

- NP-complete
- Can be solved similar to 2-SAT because the maximum sum of three variables is 1+1+1=11, which can be represented by two bits without overflow.
- The algorithm does not change, only the way to create packet is changed

Problems for parasites

- Several computational cycles are taken to process the possible solutions
- Possibility of false negatives
- Possibility of false positives
- Ethical dilemma

Problems for servers

- Delays due to processing the parasitic messages could cause a denial of service
- Almost impossible to prevent someone from running a parasitic job on your server
  - Removing or changing the exploited functions would cause the server to be unable to communicate on the Internet

References

- Barabasi et.al., *Parasitic Computing,* NATURE 412, 30 Aug 2001.
- Barabasi et.al. *Supplement material for Parasitic Computing :* http://www.nd.edu/~parasite/
- Barger N. Robert & Crowell R. Charles,*The ethics of Parasitic Computing,* Sept 2003 : www.nd.edu/~ccrowell/Parasitic%20Computing.pdf
- Ivars Peterson, *Sneaky Calculations,* Science News 160, 17 Nov 2001.
- www.hindu.com/thehindu/2001/09/13/stories/08130001.htm

Thank You