Selected Topics in

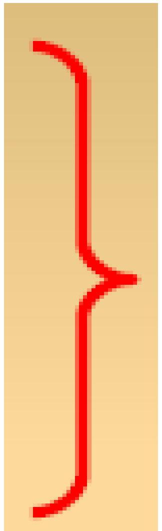
DB&Software Security

Agenda

- Buffer overflow.
- Race condition.
- SQL injection.
- Statistical database security.

Buffer overflow

- Stack overflow(covered)
- Heap overflow
- Integer overflow
- Input validation
- Return to libc
- Defenses(input validation)

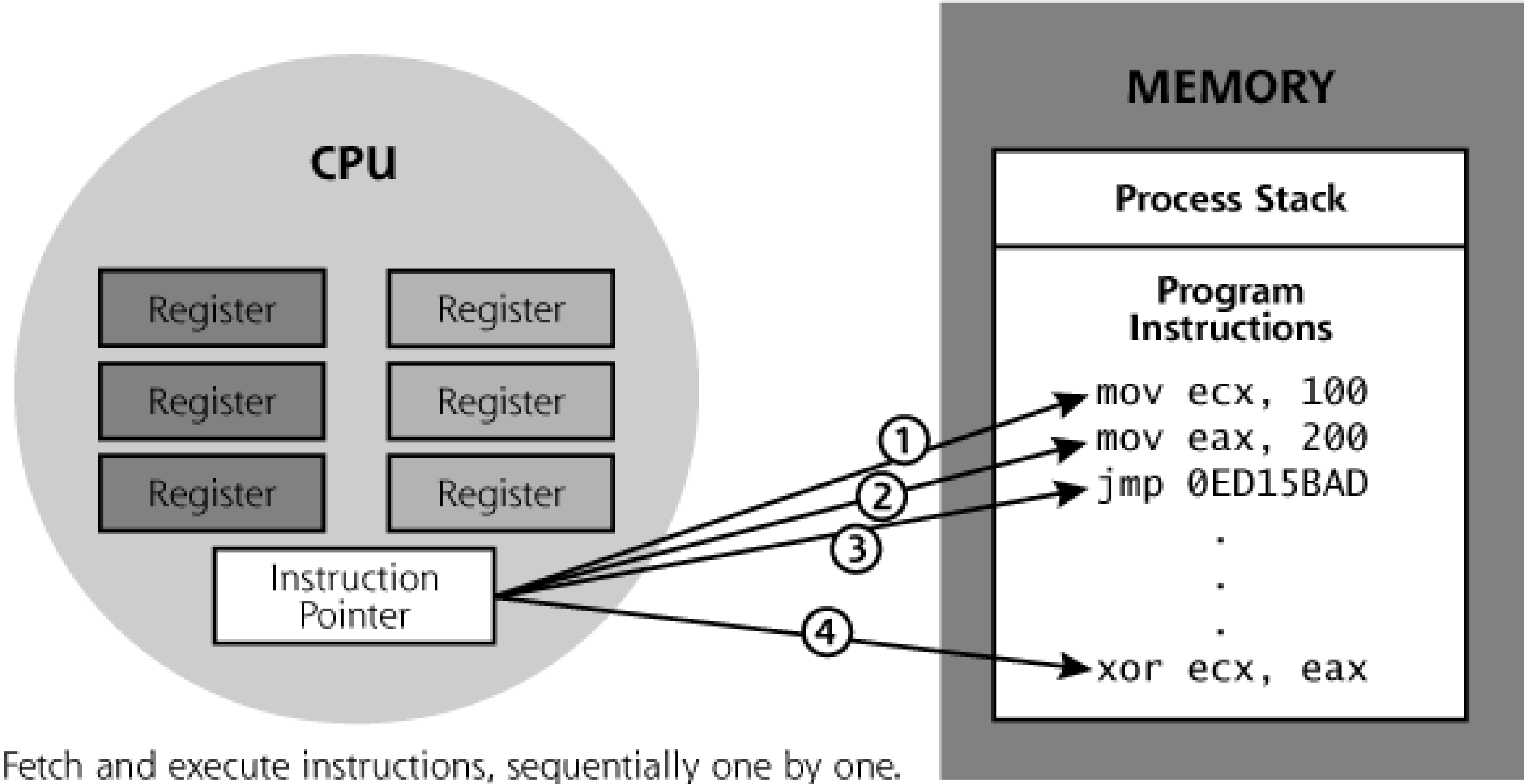


SELF STUDY

Buffer Overflow

- Buffer overflows are extremely common today, and offer an attacker a way to gain access to and have a significant degree of control over a vulnerable machine.
- “Imagine if I could execute one or two commands on your valuable server, workstation, or palmtop computer. Depending on the privileges I'd have to run these commands, I could add accounts, access a command prompt, remotely control the GUI, alter the system's configuration … anything I want to do, really. Attackers love this ability to execute commands on a target computer.”

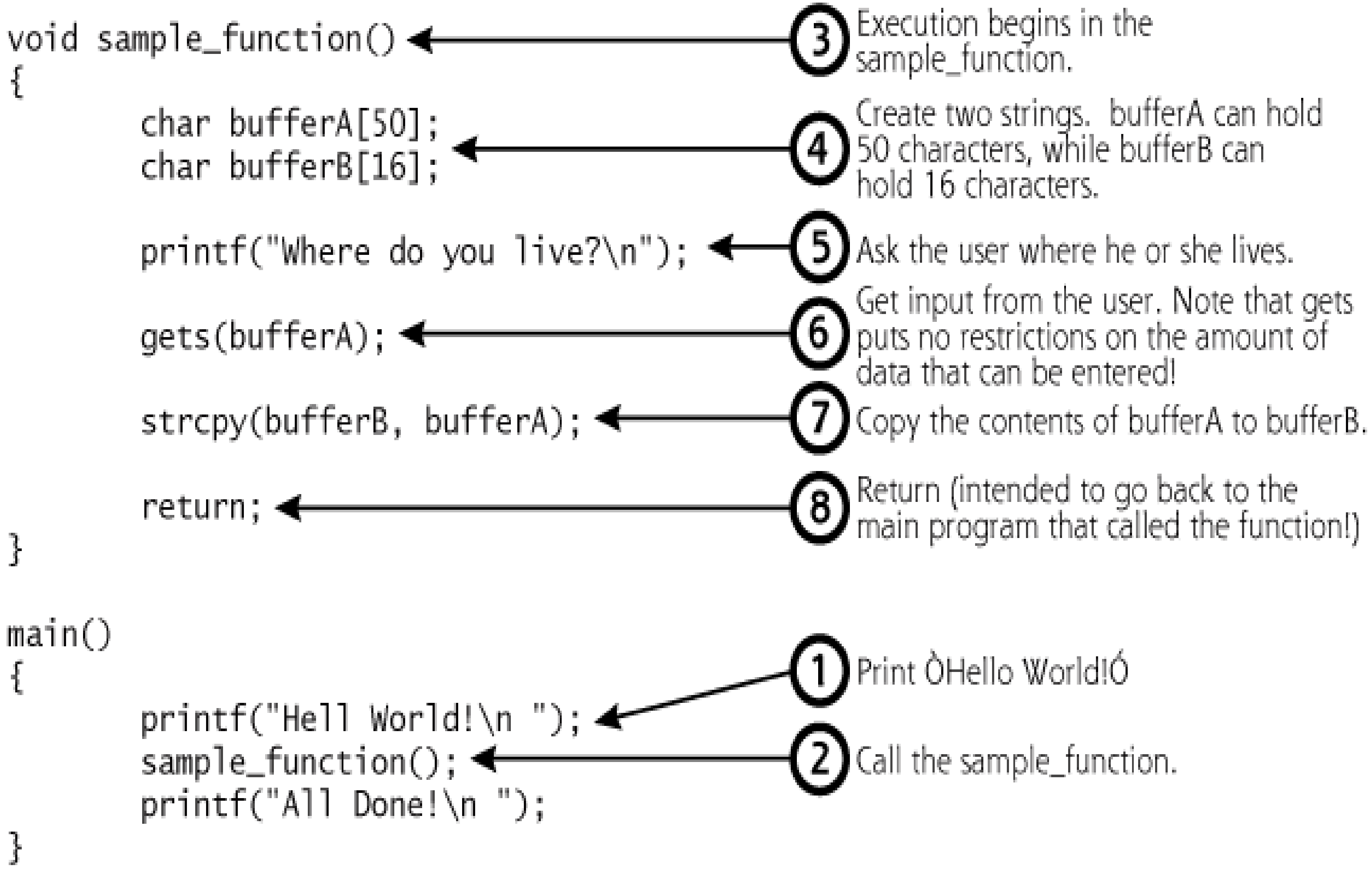
How program execute



```
mov ecx, 100
mov eax, 200
jmp 0ED15BAD
.
.
.
xor ecx, eax
```

Fetch and execute instructions, sequentially one by one.
Instruction Pointer is incremented.
At Jump, Instruction Pointer is altered to begin fetching instructions in a different location.

Vulnerable code



```
void sample_function()
{
        char bufferA[50];
        char bufferB[16];

        printf("Where do you live?\n");

        gets(bufferA);

        strcpy(bufferB, bufferA);

        return;
}

main()
{
        printf("Hell World!\n ");
        sample_function();
        printf("All Done!\n ");
}
```

3 Execution begins in the sample_function.

4 Create two strings. bufferA can hold 50 characters, while bufferB can hold 16 characters.

5 Ask the user where he or she lives.

6 Get input from the user. Note that gets puts no restrictions on the amount of data that can be entered!

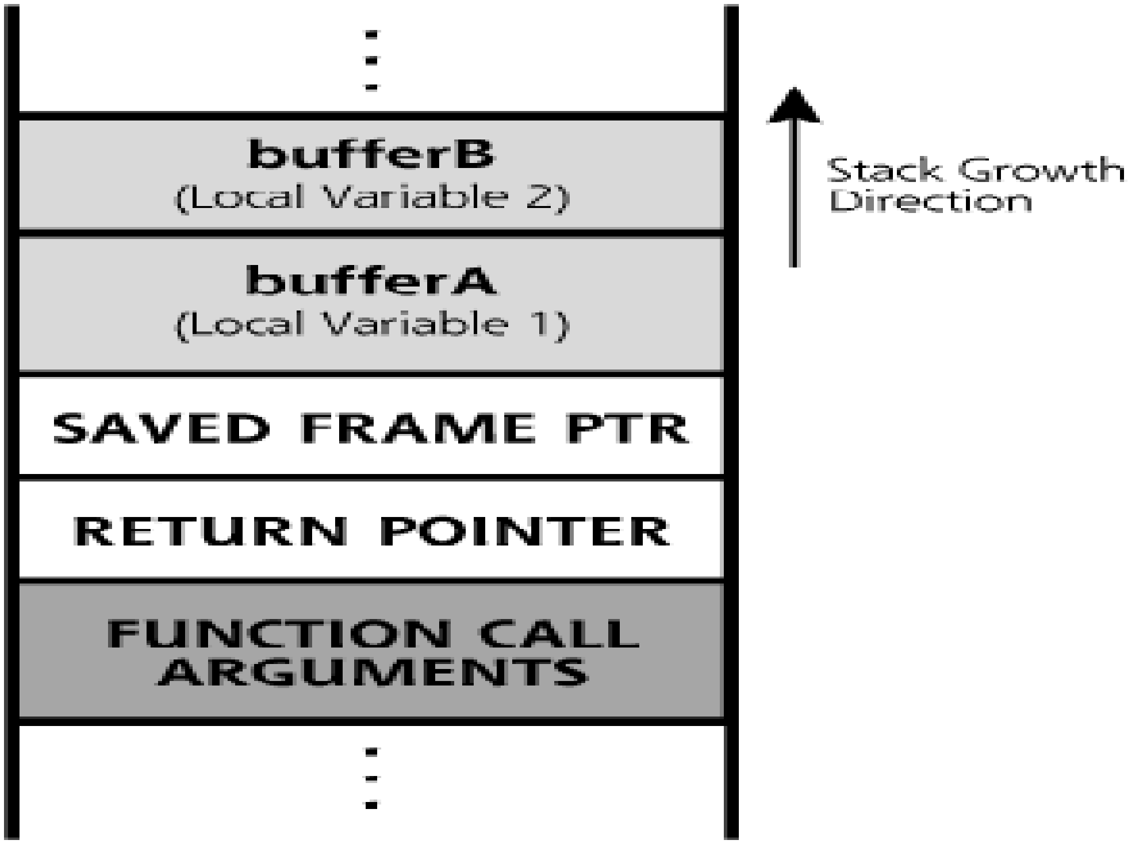
7 Copy the contents of bufferA to bufferB.

8 Return (intended to go back to the main program that called the function!)

1 Print ÒHello World!Ó

2 Call the sample_function.

Stack state

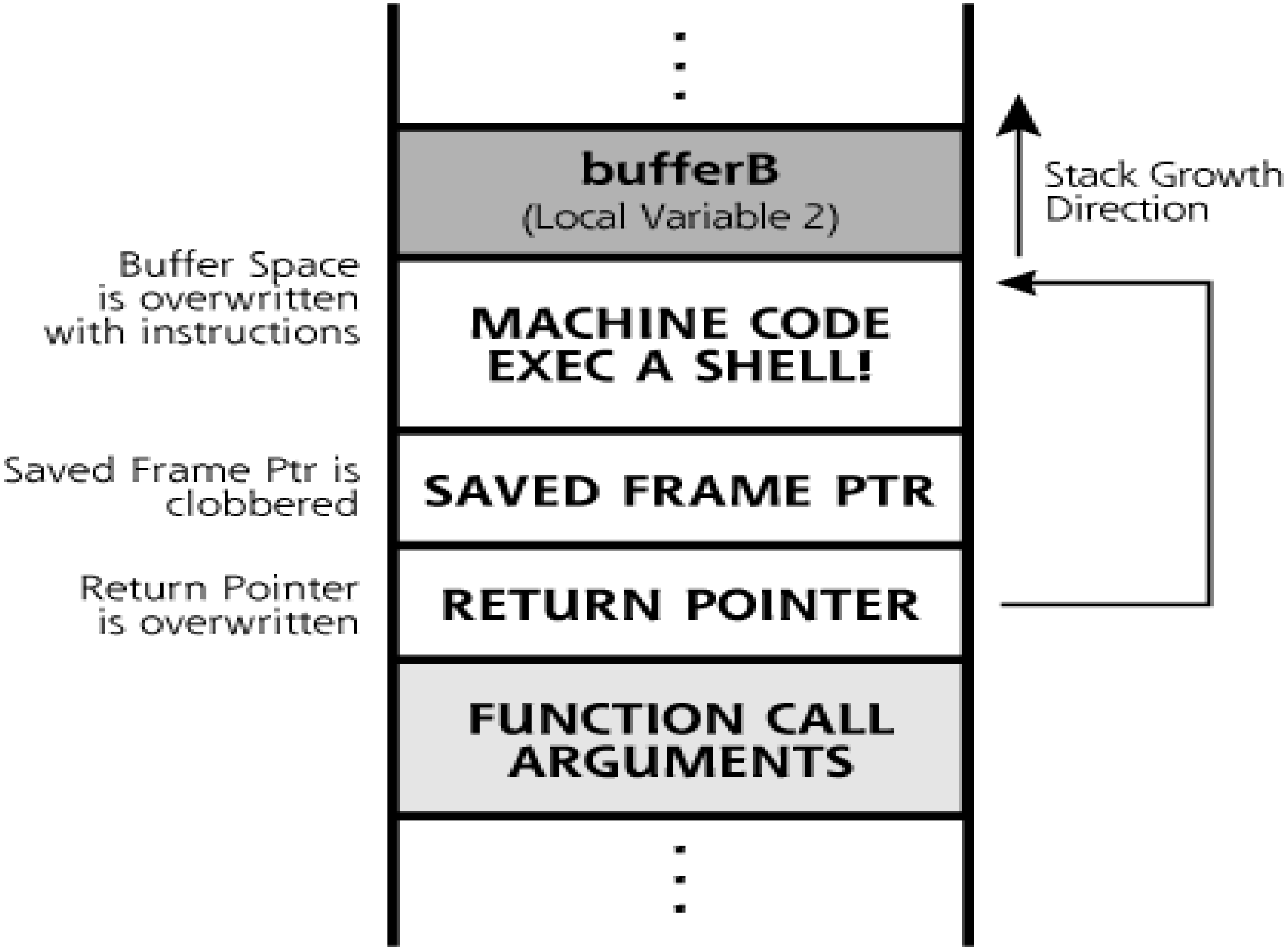


bufferB
(Local Variable 2)
bufferA
(Local Variable 1)
SAVED FRAME PTR
RETURN POINTER
FUNCTION CALL ARGUMENTS
Stack Growth Direction

Stack overflow

- The most valuable information on the stack is the so-called return address
- "At what address to execute code when the current function is ending/returning."
- Plain stack overflow
- Locate a buffer that can be overwritten, and that overwrites the return address from the function call
- Add executable code in memory to jump to when the function returns
- Overwrite the return address with the new address of our own executable code

STACK OVERFLOW -CONT



How to find

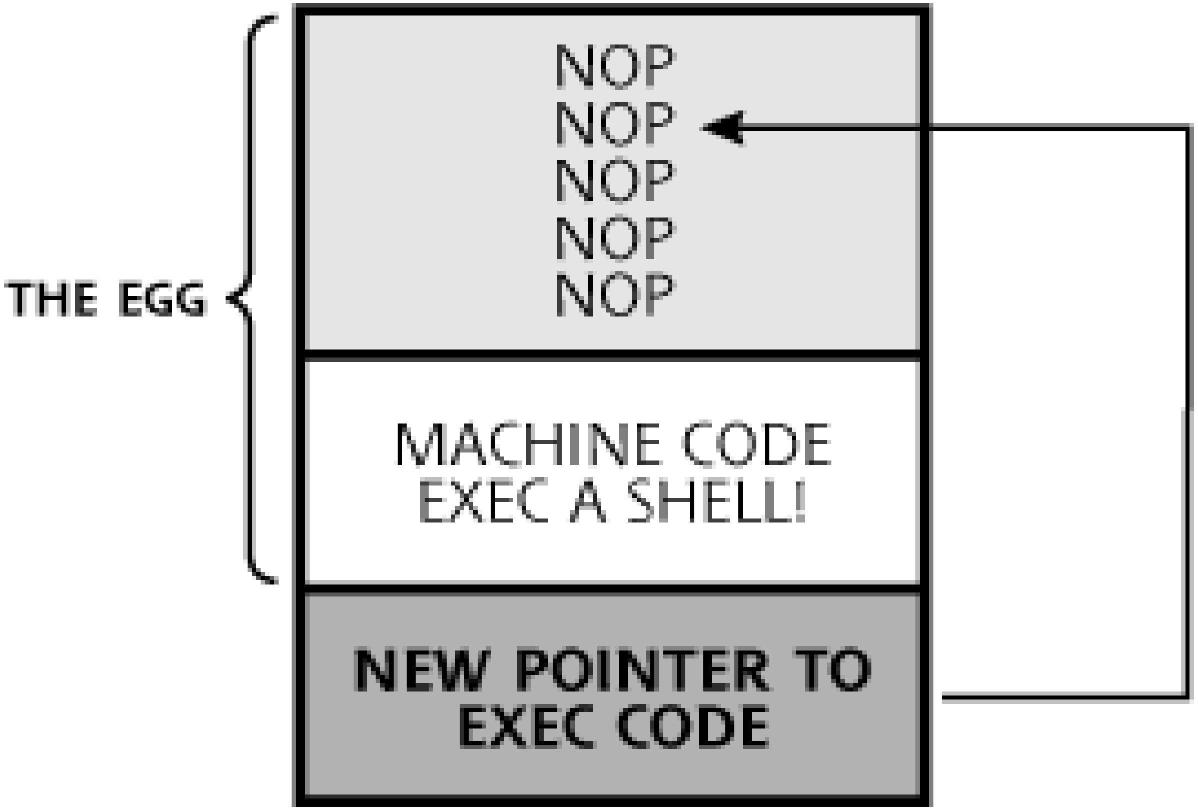
- fgets
- gets
- getws
- sprintf
- strcat
- strcpy
- strncpy
- scanf
- memcpy
- memmove

Or more brute force approach

Try the program with many different input (as much as ypu will discover a vulnerability)

```
EAX = 00F7FCC8 EBX = 00F41130 ECX = 41414141
 EDX = 77F9485A ESI = 00F7FCC0 EDI = 00F7FCC0
EIP = 41414141    ESP = 00F4106C EBP = 00F4108C
 EFL = 00000246
```

THE STRUCTURE OF AN EXPLOIT(SPLOIT)



Race condition

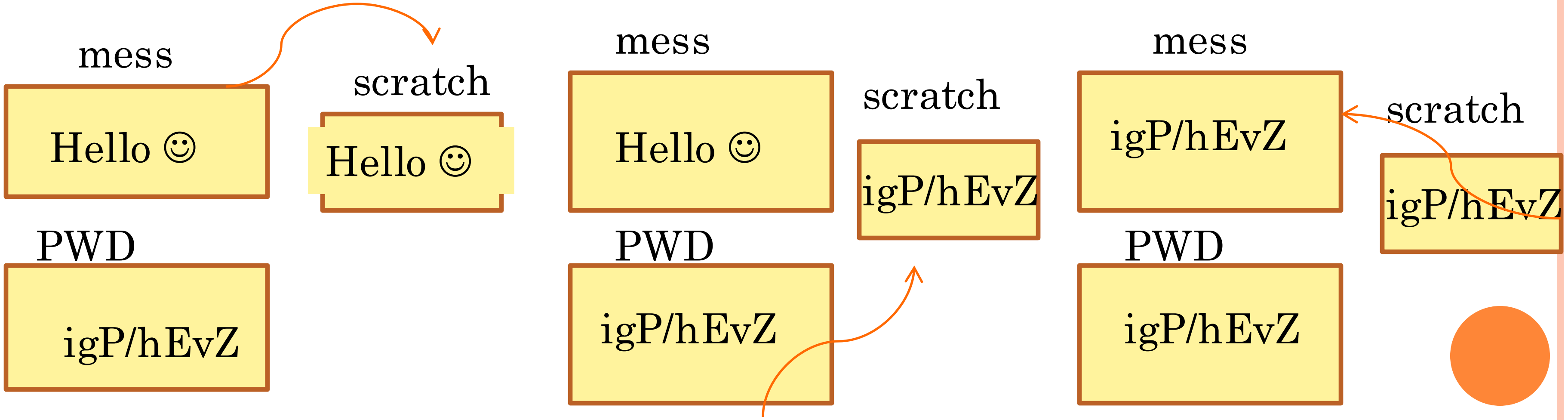
- “A race condition (or race hazard) is a flaw in a system or process whereby the output and/or result of the process is unexpectedly and critically dependent on the sequence or timing of other events. The term originates with the idea of two signals racing each other to influence the output first.”
- An attacker can try to exploit a race condition to change a value after it has been checked before it is used TOCTTOU.

EXAMPLE

- CTSS (One of the early time/sharing OS)
  - Once, a user found that the password file was given as the 'Message of the day'
  - Every user had a unique home diretory
  - When a user invoked the editor a scratch file is created which has a fixed name ,say scratch, and this name is independent of the name of the file that is desired.(This was a reasonable design decision since a user could run one application at a time and no one else has access to the user directory).so far so good ☺
  - System was treated as a user with its own directory
  - Several users are working as a system managers for certain period.

EXAMPLE-CONT

- One system manager start to edit the message of the day scratch=mess.
- Second system manager starts to edit the password file scratch=PWD.
- Then the first manager stored the edited file so we get mess=scratch=PWD.



SQL INJECTION

- Common Vulnerability

  User input used in database

  User input modified to have impact on SQL statement

- Example

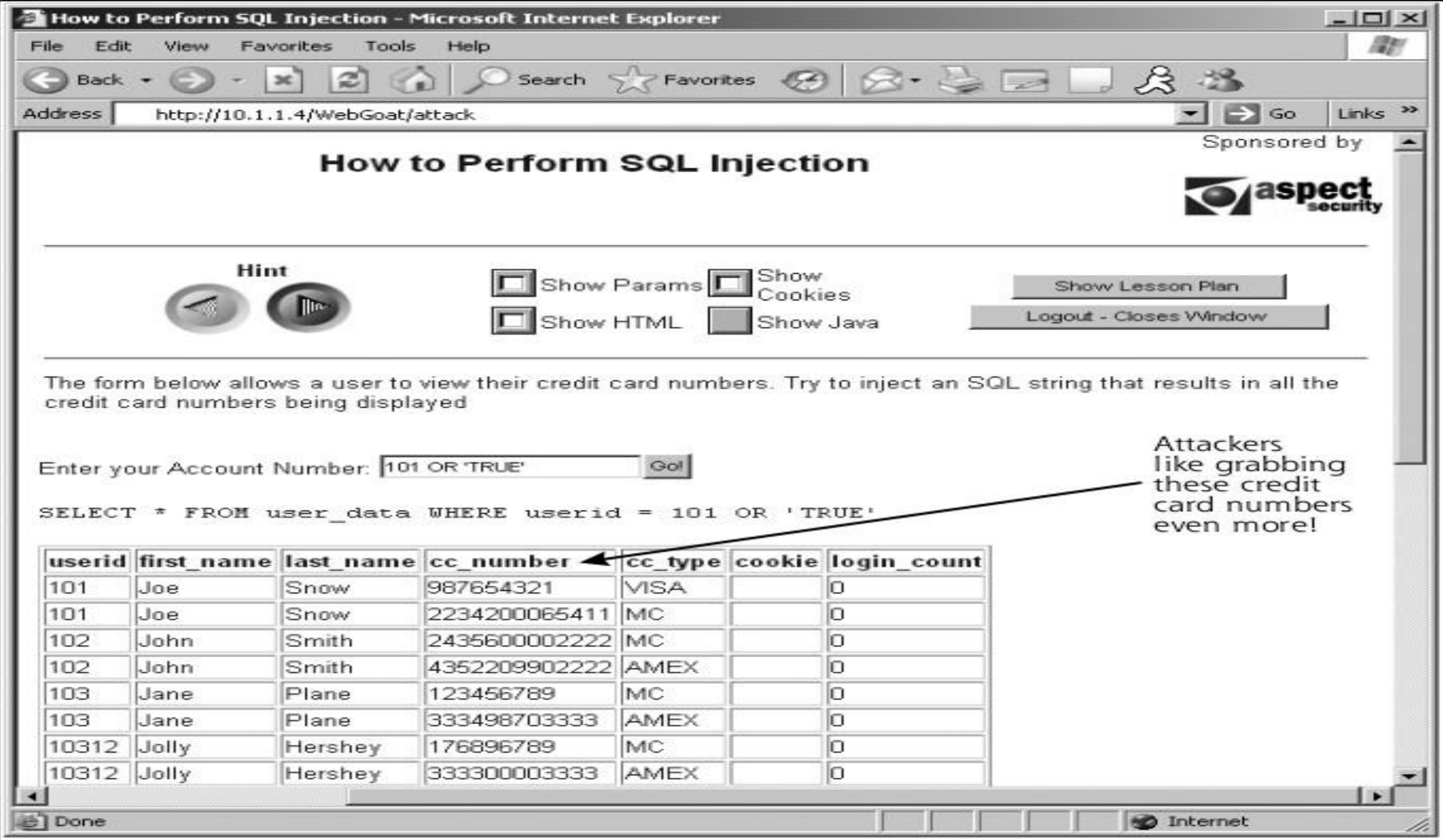
  string sql=“select * from client where name =“’”+name+” ’ ”

  The intention is to deal with select * from client where name =’Bob’

  However when an attacker enter name as

  ’Bob’ OR 1=1- -

EXAMPLE



How to Perform SQL Injection - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.1.1.4/WebGoat/attack
Go Links
Sponsored by
aspect security
Hint
Show Lesson Plan
Logout - Closes Window
Done
Internet

How to Perform SQL Injection

□ Show Params □ Show Cookies
□ Show HTML □ Show Java

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed

Enter your Account Number: 101 OR 'TRUE' Go!

```
SELECT * FROM user_data WHERE userid = 101 OR 'TRUE'
```

userid	first_name	last_name	cc_number	cc_type	cookie	login_count
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0

Attackers like grabbing these credit card numbers even more!

Figure 7.39 The evil user enters an account number of 101 or 'TRUE' to get all account information via SQL injection.

Statistical database security.

- The distinctive feature of a statistical database is that information is retrieved by means of aggregate queries
- Examples
  - COUNT
  - SUM
  - AVG
  - MAX
  - MIN

TRACKER ATTACK

- Employ statistical queries to leak sensitive information about individual.
- ”A query predicate R that allows the tracking down of information about a single tuple is called an individual tracker. A general tracker T is a predicate that can be used to find the answer to any inadmissible query.

Example

Name	Sex	Program	Units	Grade Ave
Alma	F	MBA	8	63
Bill	M	CS	15	58
Carol	F	CS	16	70
Don	M	MIS	22	75
Errol	M	CS	8	66
Flora	F	MIS	16	81
Gala	F	MBA	23	68
Homer	M	CS	7	50
Igor	M	MIS	21	70

QUERY SINGLE RECORD

- Q1

```
SELECT COUNT(*)
FROM Students
Where Sex=’F’ AND Program =’CS’
```

- Q2

```
SELECT AVG(Grade Ave)
From Students
 Where Sex=’F’ AND Program =’CS’
```

Easy to prevent with constraint query >=3

IS THE PROBLEM SOLVED

- Q1

```
SELECT COUNT(*)
FROM Students
Where Program='CS'
```

- Q2

```
SELECT COUNT(*)
FROM Students
Where Program='CS' AND Sex='M'
```

- Q3

```
SELECT  AVG(Grade Ave)
FROM Students
Where Program='CS'
```

- Q4

```
SELECT AVG(Grade Ave)
FROM Students
Where Program='CS' AND Sex='M'
```

IS THE PROBLEM SOLVED -CONT

- With a simple math
  - Q1:4
  - Q2:3
  - Q3:61
  - Q4:58

  - Carol grade = 4*61-3*58=**70 (OBS got it)**
  - **Is there any systematic way to do this or just we are lucky ☺ since Carol was the single female CS student**

So what

- Let T be the general Tracker and Let R be a predicate that identify the tuple r.
- So we can make two queries using RˇT and R ˇ Not(T). Our target r is the only tuple that is used in both queries.
  - R: Name = ’Carol’
  - T: Program=’MIS’

- Q1: SELECT SUM(Units)

```
        FROM Students
       Where Name =’Carol’ OR Program=’MIS’
```

EXAMPLE-CONT

- Q2: SELECT SUM(Units)

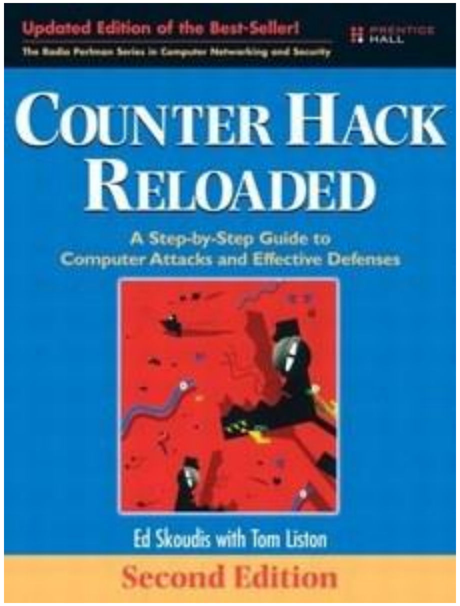
  ```
  FROM Students
  Where Name ='Carol' OR Not (Program='MIS')
  ```

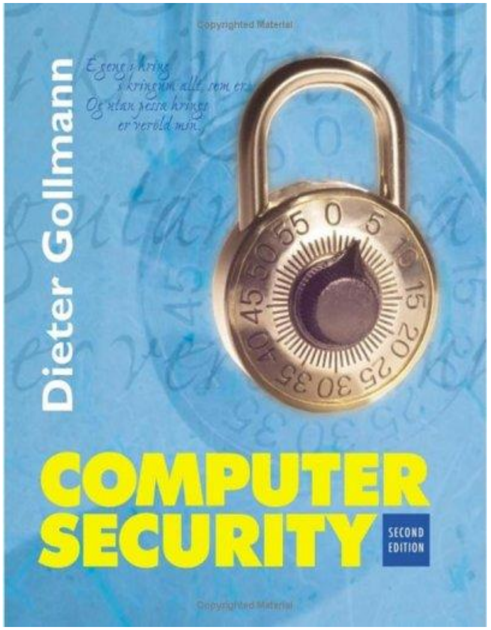
- Q3: SELECT SUM(Units)

  ```
  FROM Students
  ```

- Q1:75
- Q2:77
- Q3:136
- 136-(75+77)=16 units

BIB


Updated Edition of the Best-Seller!
The Radia Perlman Series in Computer Networking and Security
PRENTICE HALL
COUNTER HACK RELOADED
A Step-by-Step Guide to Computer Attacks and Effective Defenses
Ed Skoudis with Tom Liston
Second Edition


Dieter Gollmann
COMPUTER SECURITY
SECOND EDITION