

<RSA> Sigurnost informacijskih sistema </RSA>

<AES> predavanja </AES>

<CISSP> (Ne)Sigurnost u razvoju softvera</CISSP>



Summary

- Zašto softver?
- Buffer Overflow
- Validacija ulaza
- Malware



Lekcija 1. Zašto softver?

- Softver
- Primjeri loših rješenja
- Kompleksnost
- Gdje je potrebno obratiti pažnju?

Zašto softver?

- Zašto je softver jednako važan kao kriptografija, protokoli i sl.?
- Informacijska sigurnost je uglavnom implementirana kroz neki oblik softverskog rješenja.
- Ako je softver meta napada, tada je i sigurnost ugrožena.
- Sam softver je vrlo loša osnova za sigurnost.



Primjeri loših rješenja

- NASA Mars Lander (\$165 miliona)
 - Srušio se prilikom slijetanja...
 - ...greška u konverziji mjernih jedinica iz Amerike i Evrope
- Denver airport
 - Sistem za upravljanje prtljagom
 - prepun bug-ova
 - Odgoda otvaranja 11 mjeseci
 - Troškovi po danu \$1 milion
- ...



Kompleksnost

- “Kompleksnost je najveći neprijatelj sigurnosti”, Paul Kocher, Cryptography Research, Inc.

Sofver	Linija koda (LOK)
Netscape	17 miliona
Space Shuttle	10 miliona
Linux kernel 2.6.0	5 miliona
Windows XP	40 miliona
Mac OS X 10.4	86 miliona
Boeing 777	7 miliona

- Novo auto ima više LOK-a nego Apollo svemirske letjelice

Teme na koje je potrebno obratiti pažnju

- Programske greške

- Buffer overflow
- Validacija ulaza

- Maliciozni softver

- Virusi
- Crvi
- ...

Lekcija2. Buffer Overflow

- Scenariji
- Definicija
- Primjeri
- Validacija ulaza

Mogući scenariji

- Korisnik unosi podatke preko web forme
- Podaci se šalju na server
- Server upisuje podatke u niz koji se zove buffer, bez provjere dužine i opsega
- Podaci "prepisuju" buffer
 - Ovakve situacije mogu uzrokovati napad
 - U tome slučaju napad može izvest svako ko ima internet pristup

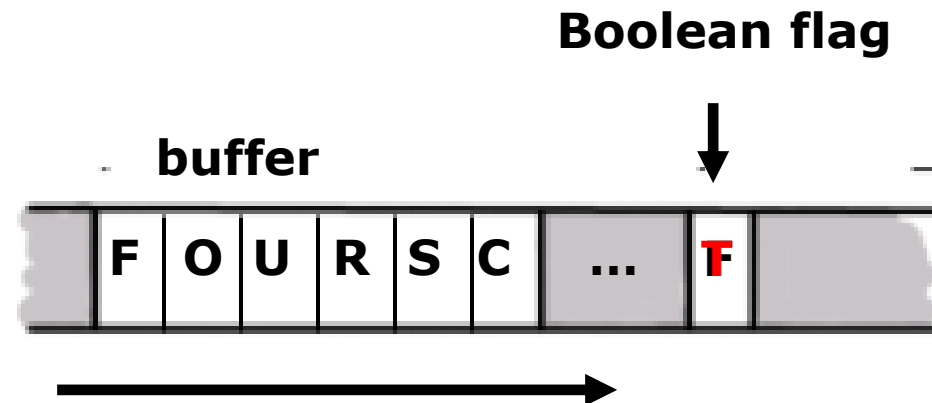
Definicija

```
int main() {  
    int buffer[10];  
    buffer[20] = 37;}
```

- Šta se desi kada se izvrši ovaj kod?
- Zavisi od toga šta se nalazi u memoriji na lokaciji "buffer[20]"
 - Može se prepisati korisnički kod ili podaci
 - Može se prepisati sistemski kod ili podaci
 - Sve može raditi kao da se ništa nije desilo.

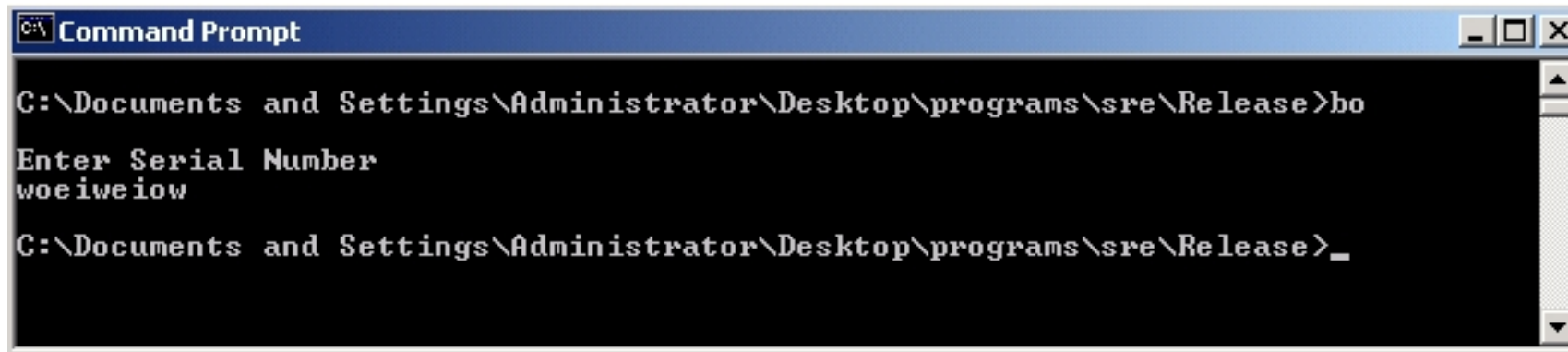
Jednostavan primjer

- Neka je *boolean flag* oznaka za uspješnost autentifikacije
- Buffer overflow može prepisati *flag* dozvoljavajući da se bilo ko autentificira



Konkretan primjer „Buffer Overflow“

- Program traži unošenje serijskog broja koji nije poznat napadaču
- Napadač **nema** source code
- Napadač ima izvršni (exe) fajl

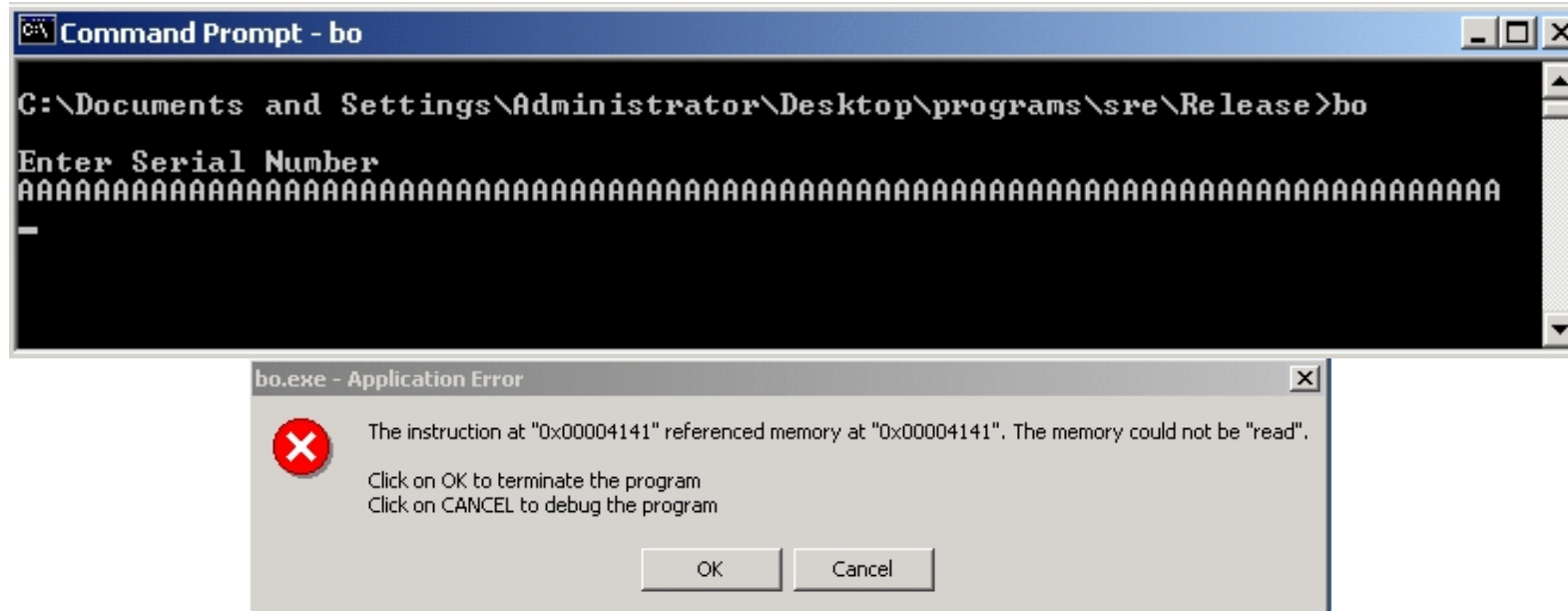


```
Command Prompt
C:\Documents and Settings\Administrator\Desktop\programs\sre\Release>bo
Enter Serial Number
woeiweiw
C:\Documents and Settings\Administrator\Desktop\programs\sre\Release>_
```

- ☐ Program prekida rad nakon unošenja neispravnog podatka

Konkretan primjer „Buffer Overflow“

- Na osnovu greške napadač otkriva da postoji opcija prepisivanja buffer-a



- Primjetite da je 0x41 ASCII kod za "A"

Deasenbliranje koda

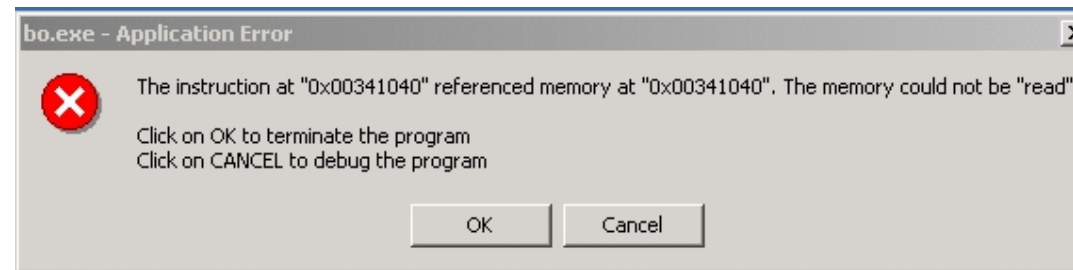
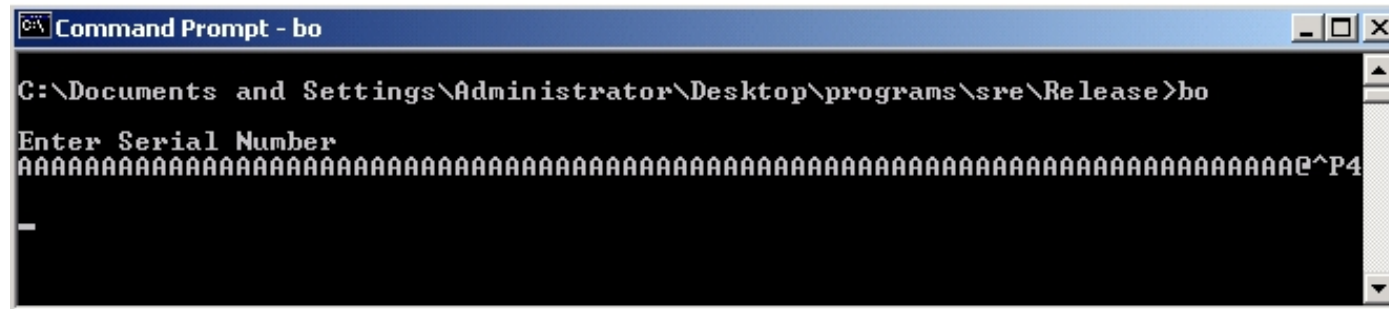
```
.text:00401000
.text:00401000
.text:00401003
.text:00401008
.text:0040100D
.text:00401011
.text:00401012
.text:00401017
.text:0040101C
.text:0040101E
.text:00401022
.text:00401027
.text:00401028
.text:0040102D
.text:00401030
.text:00401032
.text:00401034
.text:00401039
.text:0040103E

sub     esp, 1Ch
push    offset aEnterSerialNum ; "\nEnter Serial Number\n"
call    sub_40109F
lea     eax, [esp+20h+var_1C]
push    eax
push    offset aS              ; "%5"
call    sub_401088
push    8
lea     ecx, [esp+2Ch+var_1C]
push    offset aS123n456 ; "S123N456"
push    ecx
call    sub_401050
add     esp, 18h
test    eax, eax
jnz     short loc_401041
push    offset aSerialNumberIs ; "Serial number is correct.\n"
call    sub_40109F
add     esp, 4
```

- ❑ Cilje je prepisati adresu 0x401034

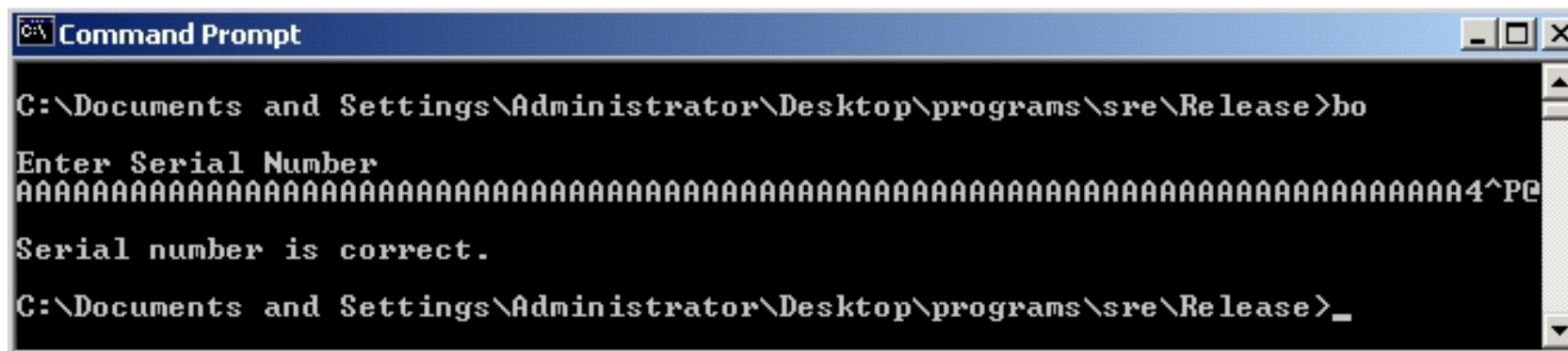
Konkretan primjer „Buffer Overflow“

- U ASCII, 0x401034 je "@^P4"



Konkretan primjer „Buffer Overflow“

- Ako okrenemo "4^P@"



```
Command Prompt
C:\Documents and Settings\Administrator\Desktop\programs\sre\Release>bo
Enter Serial Number
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4^P@
Serial number is correct.
C:\Documents and Settings\Administrator\Desktop\programs\sre\Release>_
```

- ❑ Uspjeli smo i zaobišli unošenje ispravnog serijskog broja.

Source Code

```
#include <stdio.h>
#include <string.h>

main()
{
    char in[75];

    printf("\nEnter Serial Number\n");

    scanf("%s", in);

    if(!strcmp(in, "S123N456", 8))
    {
        printf("Serial number is correct.\n");
    }
}
```



Validacija ulaza

- Web forma
- Validacija se vrši na klijentu
- Npr.

`http://www.things.com/orders/final&custID=112&num=55A&qty=20&price=10&shipping=5&total=205`

- Ulaz nije provjeren na serveru
 - Navedeni URL može izgledati ovako

`http://www.things.com/orders/final&custID=112&num=55A&qty=20&price=10&shipping=5&total=25`

Lekcija 3. Malware

- Maliciozni softver
- Gdje virusi „žive“
- Hall of fame
 - Code Red (2001)
 - SQL Slammer (2004)
- Detekcija
- Budućnost malicioznog softvera?

Maliciozni softver

- Virusi nisu ništa novo...
 - Prvi datiraju sa početka 80'tih
- Vrste (mada postoje preklapanja)
 - Virusi - pasivno širenje
 - Crvi - aktivno širenje
 - Trojanski konji - neočekivane funkcionalnosti
 - Trapdoor/backdoor - neovlašteni pristup
 - Rabbit - iscrpljivanje sistemskih resursa



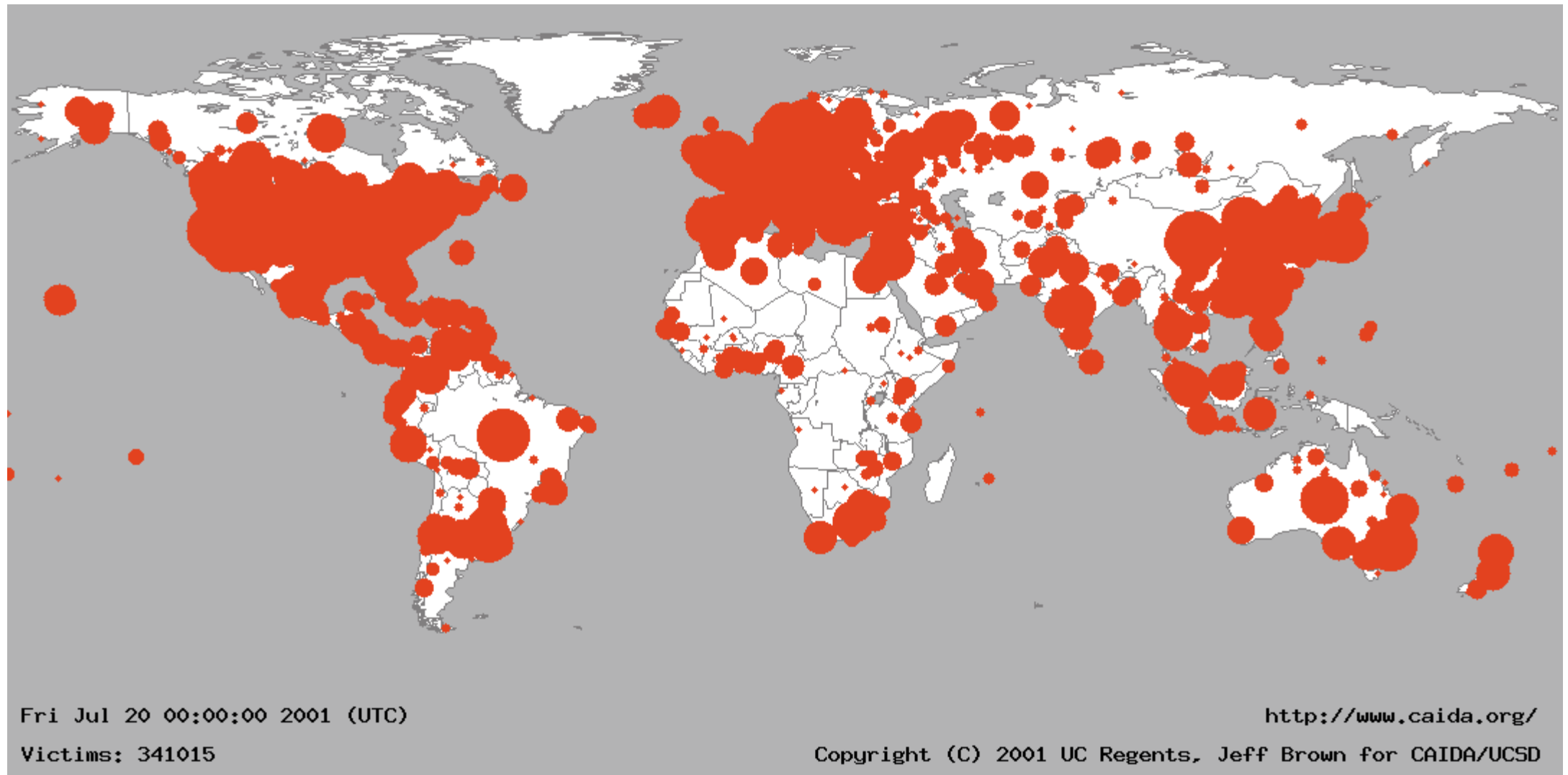
Gdje virusi „žive“?

- Boot sektor
 - Uzimaju kontrolu nad sistemom prije svih operacija
- Rezidentni u memoriji
- Aplikacije, makro funkcije ...
- Kompajleri, debugari.

Code Red Worm

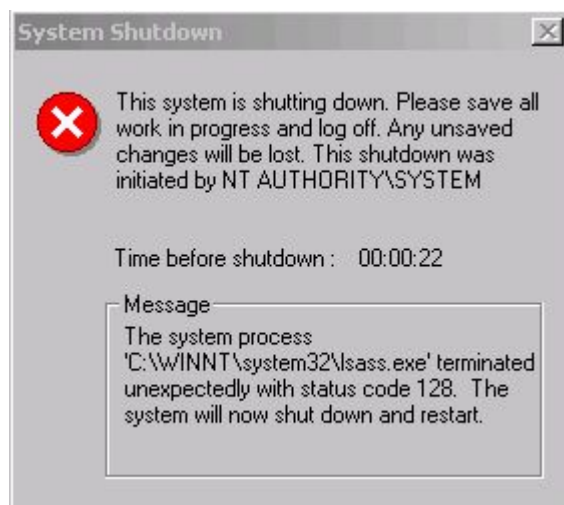
- Pojavio se u julu 2001
- Inficirao je više od 250,000 sistema unutar 15 sati
- Kasnije još 750,000 od približno 6,000,000 ranjivih sistema
- Dan 1 do 19 širenje na što više sistema
- Dan 20 to 27: (DDoS) na www.whitehouse.gov
- Neki su govorili da je ovo bilo beta testiranje informatičkog ratovanje - Cyberwarfare
 - Naravno, ovo niko nije potvrdio, ali nije niti demantovao

Code Red Worm

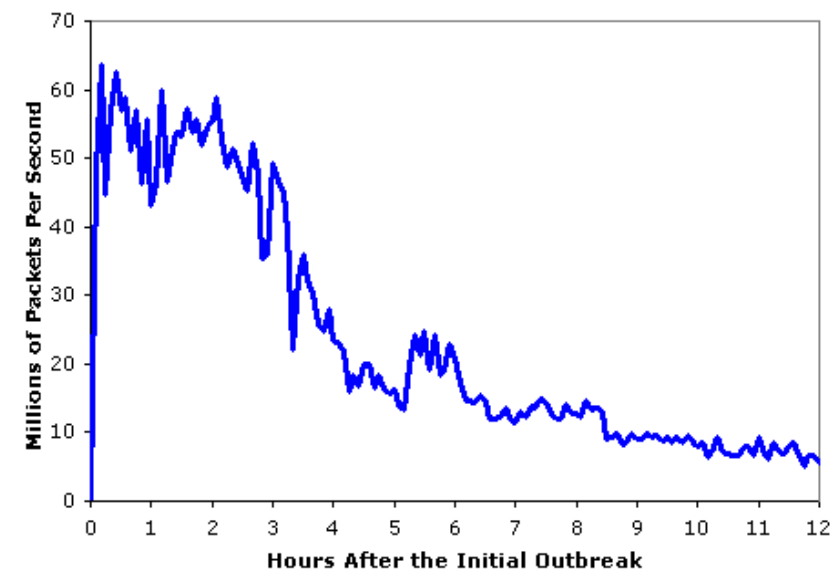


SQL Slammer

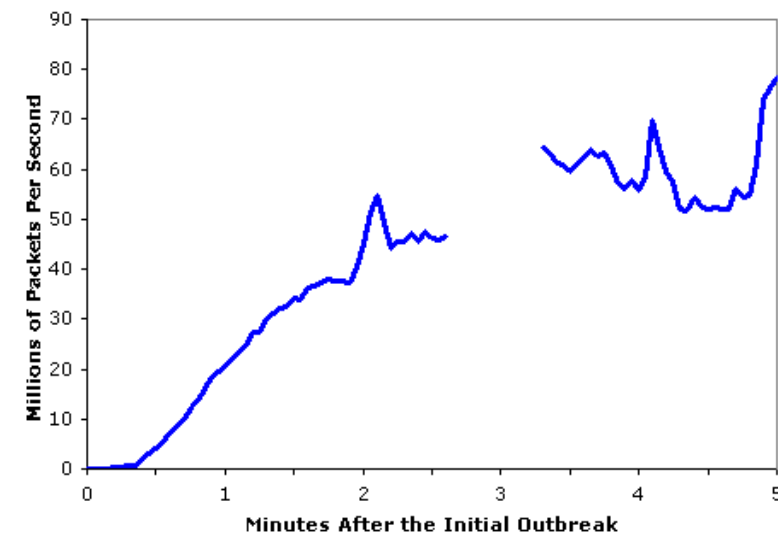
- Inficirao **75 000 sistema u 10 minuta**
- Na vrhuncu, infekcija se udvostručavala svakih 8.5 sekundi



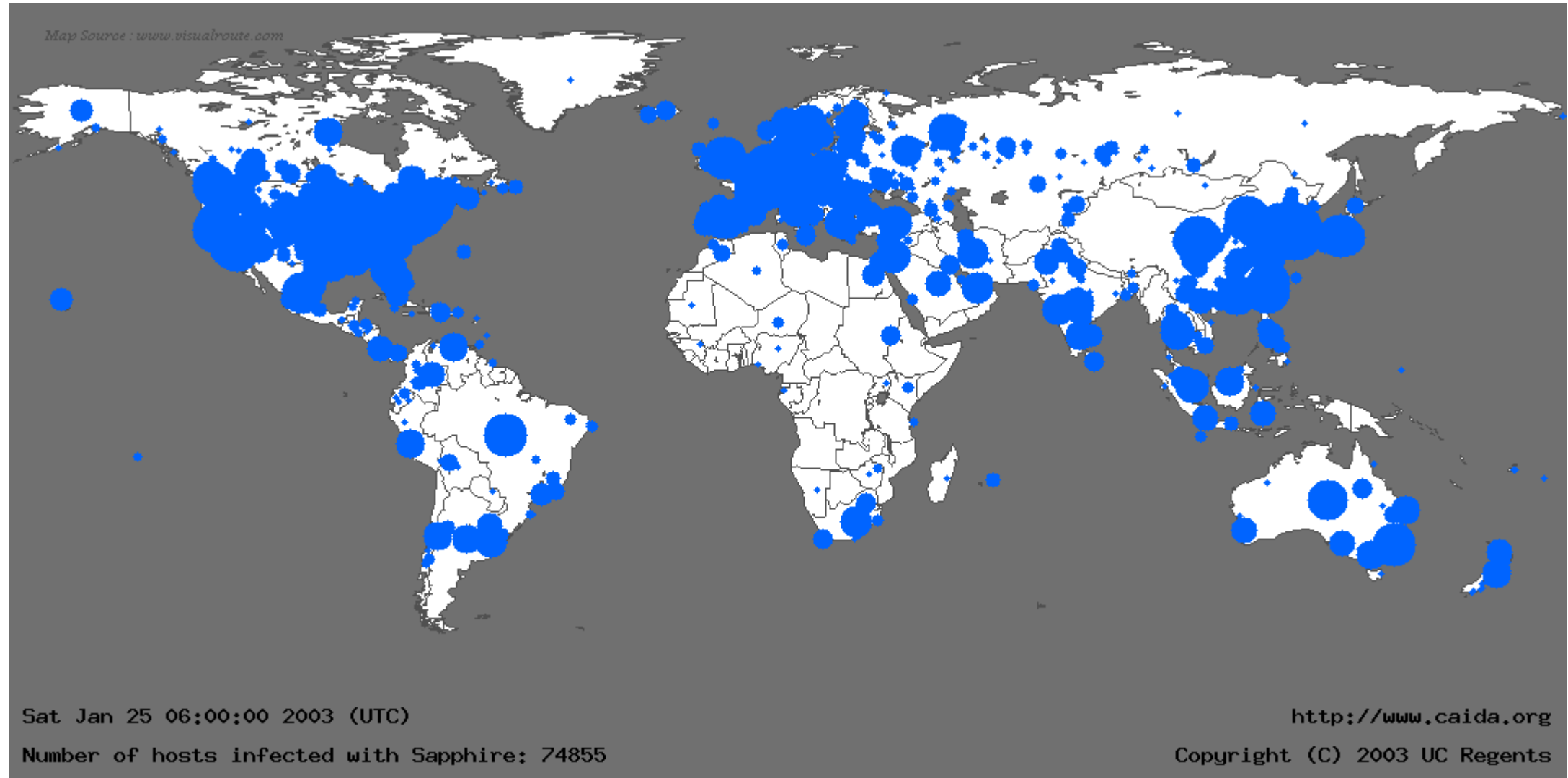
Aggregate Scans/Second in the 12 Hours After the Initial Outbreak



Aggregate Scans/Second in the first 5 minutes based on Incoming Connections To the WAIL Tarpit



SQL Slammer



Detekcija malicioznog softvera

- Signature detection
- Change detection
- Anomaly detection

Detekcija promjene - Change Detection

- Virusi se uvijek nalaze na nekoj lokaciju
- Ako se detektuje promjena u samom fajlu..možda je virus
 - Hash files
 - Periodično provjeriti hash-ove
 - Ako se promjene, fajl je možda inficiran

Budućnost malicioznog softvera

- Encrypted, polymorphic, metamorphic malware
- Fast replication/Warhol worms
- Flash worms, slow worms
- Botnets

Pitanja

