Security 101

Guido Bolognesi
guido@kill-9.it
http://www.kill-9.it

Presentazione

- Relatore
- Partecipanti
- Mini sondaggio

CIA

- La security in tre lettere
  (no, non é la CIA a cui pensate voi)

- C - confidentiality
- I - integrity
- A - availability

Confidentiality

- Assicura la privacy (nessun accesso a chi non ha diritto)
- Si applica sia ai dati sui dischi che a quelli in transito sulla rete
- Autenticazione, controllo degli accessi

Integrity

- Si applica ai dati sui dischi e a quelli in transito sulla rete
- Permette di sviluppare “fiducia” nei sistemi e nella rete di comunicazioni
- Non deve essere possibile modificare i dati in transito, qualunque sia il percorso e in qualunque modo

Availability

- Un altro catalizzatore di “fiducia”

- Necessaria, per tutti i dati

- La disponibilità dei dati é fondamentale. Senza avere i dati, le altre caratteristiche diventano poco rilevanti. (DOS)

Un altro luogo comune

- Più viene introdotta “security”, meno é semplice utilizzare i sistemi

- user=user, password=blank

- user=user, one time password (alfanumerica, con segni di interpunzione, da 10 caratteri), controllo biometrico dell'iride, connessione cifrata, accesso a tempo

Un concetto chiave

Raise
The
bar

La security é un processo

Read bugtraq every day.
New vulnerability.
Rush to plug it up.

Cosa mi può succedere?

- Un po' di tutto
- Le cose più probabili?
  - traffico inaspettato dalla macchina (che venga usata come testa di ponte)
  - Processi ignoti
  - Nuovi utenti
  - Log o altri file alterati, cancellati

A cosa devo stare attento?

- Errori di configurazione o configurazioni di default non sicure
- Utenti di default
- Servizi attivi e non richiesti
- Vulnerabilità note, ce ne sono su un grande numero di software e kernel

Tipo?

- OpenSSL
- SSH
- Apache
- IIS o Windows in genere senza patch
- SQL Server
- PhpNuke
- ...

Qualche numero

Incidenti di security denunciati

1991 - 406

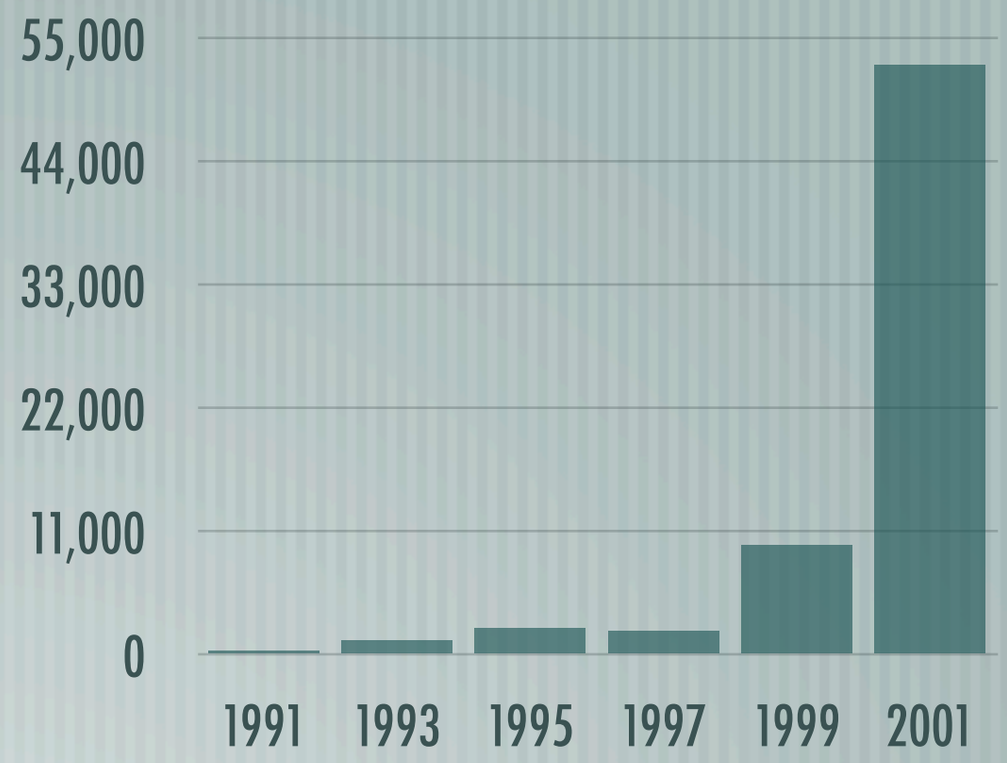
1993 - 1,334

1995 - 2,412

1997 - 2,134

1999 - 9,859

2001 - 52,658



Mediamente, il 37%
di quelli avvenuti

Tecniche

- Sniffing
- Session Hijacking
- Configurazioni
- Buffer overflow
- Problemi applicativi
- Broken standards
- Social engineering

Sniffing

- Shared (*)
- Switched (**)
- Switched tunneled+ (***)
- Wireless (**)

Session Hijacking

Mitnick,
Shimomura
&&
Toad.com

Cattive configurazioni

- Ftp anonimo in scrittura
- C$ in share
- Sendmail execution
- / come httproot
- oracle/oracle
- QSECOFR?

Buffer Overflow

- Wu-ftpd, il buco con il server intorno
- Sshd – crc32 compensation, PrivilegeSeparation
- Apache, chunked POST
- Microsoft UpNP
- Telnetd
- SNMP
- Quelli locali?

Buffer overflow

Buffer overflow
vacationing sysadmin
computer is mine

Buffer overflow

- Semplificando molto, si dividono in tre famiglie:
  - Stack based (i più comuni)
  - Heap based
  - Format string overflow

Buffer overflow

- Funzionano sovrascrivendo un'area di memoria dove si trova
- il return address (quando la funzione termina, invece che ritornare da dove é arrivata l'esecuzione va altrove)
- un function pointer (quando viene invocata la funzione, viene eseguito altro)

Buffer overflow

Un testo di esempio per tutti:

"Smashing the Stack for Fun and Profit"

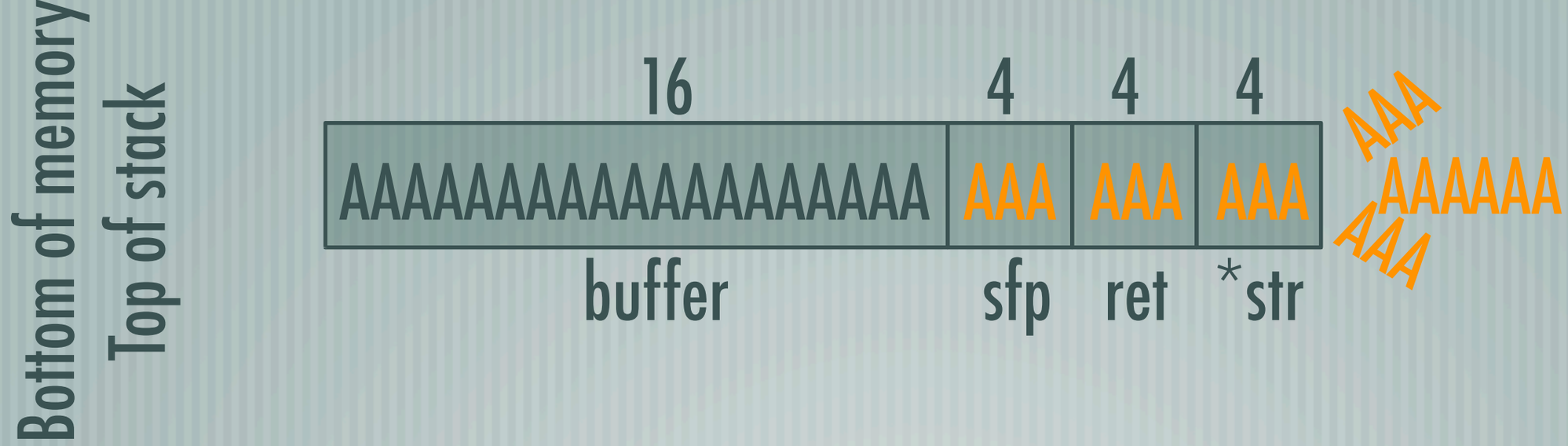
di Aleph One

pubblicato in Phrack,

volume 7, numero 49

Buffer overflow

- Proviamo un esempio “visuale”



- il return address diventa 0x41414141...

Considerazioni

- strcpy(), gets()
- Tutti i byte NULL dell'exploit devono essere convertiti (si può usare xor)
- Il return address va sovrascritto con qualcosa che punti al buffer, o a una funcall che ritorni all'interno del buffer
- Come scopriamo l'indirizzo del buffer in cui ritornare? (trial and error)

Fatta la legge...

- StackGuard é un metodo che protegge dai buffer overflow inserendo una "canary word" prima dell'indirizzo di ritorno.

- Domanda: come scrivo un exploit per una applicazione compilata con StackGuard?

Vita reale: Blaster

- Buffer overflow di Microsoft DCOM RPC (tcp/135)
- Scarica sull'host infetto mblast.exe utilizzando TFTP e lo esegue
- mblast.exe tenta un synflood verso windowsupdate.com e tenta di infettare altri host (40% locali, 60% remoti)

Vita reale: Blaster

- Alcuni “errori” nello sviluppo:
- Utilizza TCP (latency bound)
- windowsupdate.com era un alias per windowsupdate.microsoft.com, quindi a M$ é stato sufficiente rimuoverlo

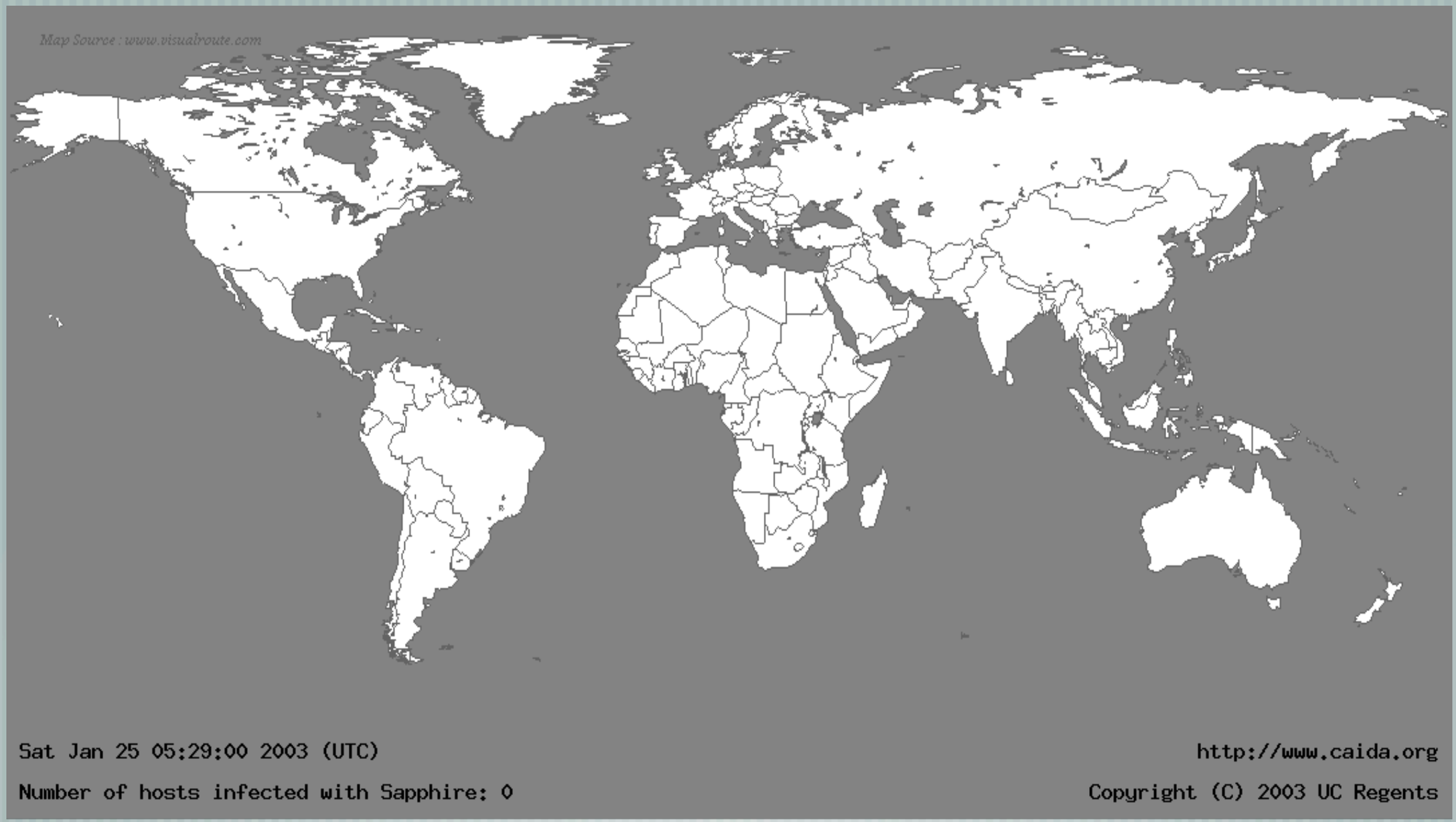
Vita reale: Slammer/Sapphire

- Primo esempio di un worm ad alta velocità
- 75.000 macchine in 30 minuti
- 90% degli host vulnerabili in 10 minuti
- Buffer overflow di un bug in SQL Server, per cui c'era una patch da 6 mesi

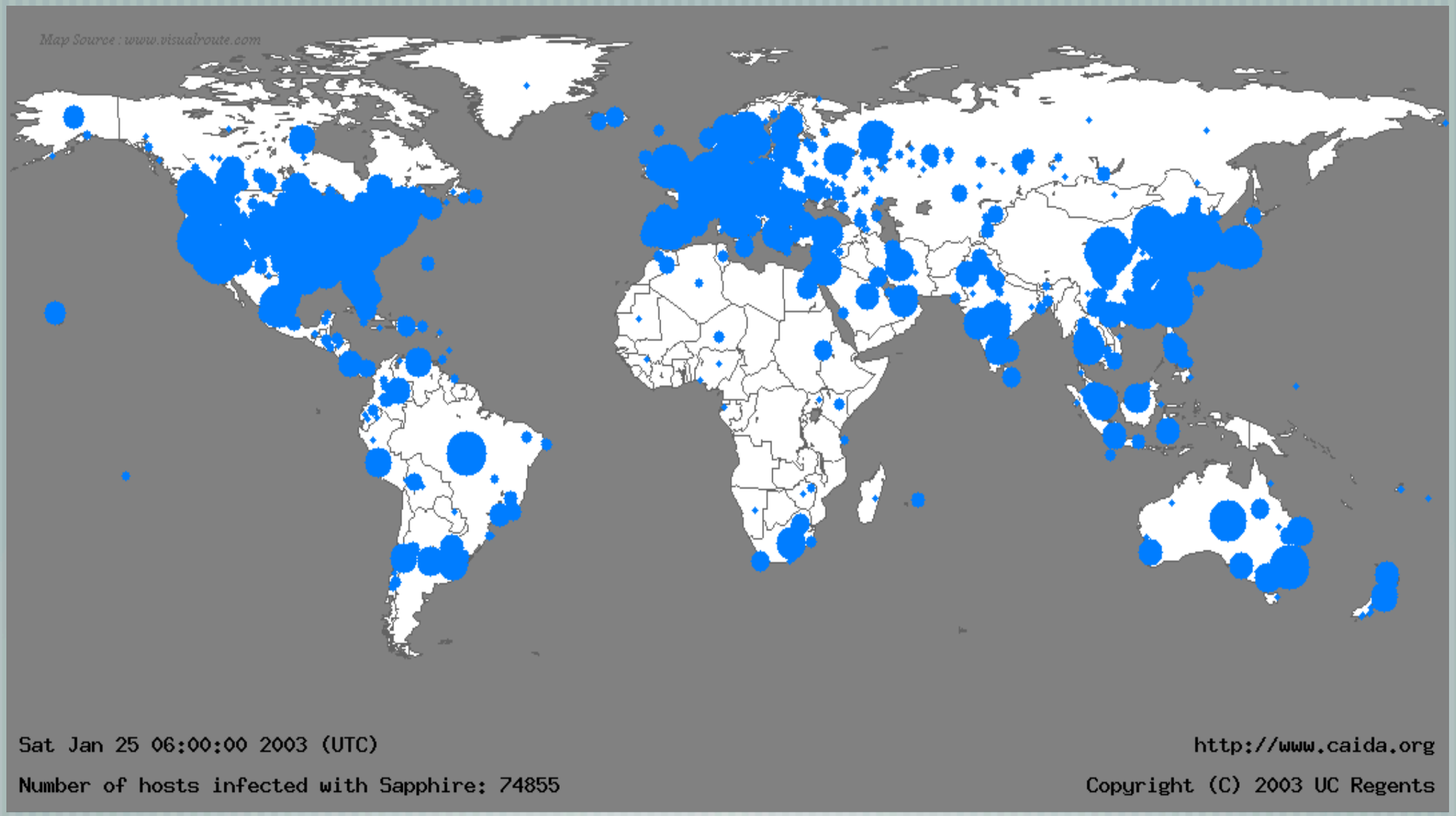
Vita reale: Slammer

- Genera un indirizzo “randomico” verso il quale propagarsi
- Utilizza UDP (banda piena)
- 376 byte di payload
- Tempo di raddoppio, circa 8.5 secondi
- Rate massimo di scan (55 milioni di scan al secondo) raggiunto in 3 minuti

Slammer da vedere: 5:29'



Slammer da vedere: 6:00'



Slammer, come funziona

- Se arriva un pacchetto UDP sulla porta 1434, con il primo byte a 0x04, il resto del pacchetto viene interpretato come una chiave di registro da aprire
- Il nome della chiave (il resto del pacchetto) viene salvato in un buffer da usare dopo
- I limiti del buffer non vengono controllati. Se la stringa é troppo lunga... inizia il divertimento.

Problemi applicativi

- /scripts/root.exe?/c+dir
- /..%c0%af../
- Phpnuke
- filename.cgi%00
- ' or 1=1
- Dns poisoning
- SNMP

Broken Standards

e-mail attachment
installs computer virus
scripting can be fun

Social Engineering

Root password mumbled
aloud in a crowded lab
how fast can I type?

(D)DOS

Packets on a wire
consuming all my bandwidth
d-o-s victim

Come mi proteggo?

In tanti modi

- Con un sistema operativo decente
- Con software aggiornato
- Rimanendo informati
- Cercando di capire come funzionano veramente le cose
- E con le solite cose noiose (backup, password...)

Cosa devo desiderare?

- Un bello stack TCP/IP (IPID, source port, TTL ~ random)
- Codice scritto bene: mktemp(), tmpnam() vs. mkstemp(), tempnam()
- Random inodes, link simbolici/hard
- Base pointer dello stack, base della mmap() ...
- PID random...

Eh... e poi?

- Grsecurity, SELinux
- TrustedBSD, OpenBSD
- Chroot, systrace, jail
- Virtual server
- Hardening generico delle macchine (Solaris/AIX/Windows...)
- ...tanta fantasia, tenendo presente che le macchine vanno anche amministrate, ed utilizzate di solito

E lato network?

- Esistono molte blackbox per network security
- Molte sono poco più che macchine generiche adattate a funzioni specifiche
- Alcuni sono oggetti estremamente specializzati (architettura, software ed hardware)

Lato network: esempi

- Antivirus (sia per traffico SMTP, sia per HTTP)
- Firewall (con livelli variabili di “deep inspection” e comprensione dei protocolli)
- Cache
- IDS/IPS
- Router
- Monitoraggio e log handling

Lato network: esempi

Antivirus

- oltrepassando le utopie, una rete moderna ospita anche macchine Windows
- sicuramente le più bersagliate (e “male utilizzate”)
- di conseguenza le più pericolose
- un antivirus protegge da exploit via mail e via http

Lato network: esempi

Firewall

- sicuramente uno degli oggetti più comuni
- probabilmente uno di quelli peggio configurati, mediamente
- disponibili in molti gusti (ha/bilanciamento, filtering da layer2 fino a layer7)
- ...con fasce di prezzo estremamente variabili
- ...difficili da valutare correttamente

Lato network: esempi

Cache

- siamo certi che capiscano veramente il protocollo
- offrono molteplice protezioni
- filtrano il traffico web in ingresso ed in uscita
- bloccano il traffico web “pericoloso”
- ottimizzano l’utilizzo di banda

Lato network: esempi

IDS/IPS

- Forse uno degli oggetti più citati negli ultimi anni
- Molto difficili da mettere in produzione
- Intrusion Detection o Prevention?
- Inline o Offline?
- Attivi o Passivi?
- basati su firme? euristica? sfera di cristallo?

Lato network: esempi

- Router
  - come mai?
  - dove non si arriva, non si fanno danni
  - non sempre serve “full routing” in un ambiente di produzione
  - ACL: banali, ma semplici ed economiche

Lato network: esempi

- Monitoraggio e log handling
  - aiuta a quantificare la latenza della rete
  - non è possibile fare troubleshooting senza conoscere i dati di base
  - aiuta a stimare il traffico e a scalare prima che ce ne sia bisogno
  - permette un unico punto di osservazione dell'ambiente, per vasto che sia
  - ...i grafici aiutano i manager. E i manager aiutano voi. :)

Grazie a:

- S. Felix Wu, “Buffer Overflows” ECS 150
- Mark Shanek, “Smashing the stack”
- Eric Pancer, “computer security 101”
- Keynote :)

Guido [Zen] Bolognesi - guido@kill-9.it