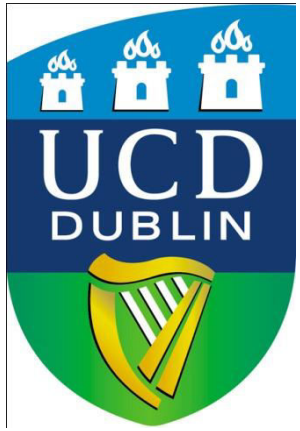
COM307000 - Software

Dr. Anca Jurcut
E-mail: anca.jurcut@ucd.ie

School of Computer Science and Informatics
University College Dublin,
Ireland



Malware

Malicious Software

- Malware is not new…
  - Fred Cohen's initial virus work in 1980's
  - Cohen used viruses to break MLS systems
- Types of malware (no standard definition)
  - **Virus** — passive propagation
  - **Worm** — active propagation
  - Trojan horse — unexpected functionality
  - Trapdoor/backdoor — unauthorized access
  - Rabbit — exhaust system resources
  - Spyware — steals info, such as passwords

Where do Viruses Live?

- They live just about anywhere, such as…
- Boot sector
  - Take control before anything else
- Memory resident
  - Stays in memory
- Applications, macros, data, etc.
- Library routines
- Compilers, debuggers, virus checker, etc.
  - These would be particularly nasty!

Malware Examples

- Brain virus (1986)
- Morris worm (1988)
- Code Red (2001)
- SQL Slammer (2004)
- Stuxnet (2010)
- Botnets (currently fashionable malware)
- Future of malware?

Brain

- First appeared in 1986
- More annoying than harmful
- A prototype for later viruses
- Not much reaction by users
- What it did
  1. Placed itself in boot sector (and other places)
  2. Screened disk calls to avoid detection
  3. Each disk read, checked boot sector to see if boot sector infected; if not, goto 1
- Brain did nothing really malicious

Morris Worm

- First appeared in 1988
- What it tried to do
  - Determine where it could spread, then…
  - …spread its infection and…
  - …remain undiscovered
- Morris claimed his worm had a bug!
  - It tried to re-infect infected systems
  - Led to resource exhaustion
  - Effect was like a so-called rabbit

How Morris Worm Spread

- Obtained access to machines by…
  - User account **password guessing**
  - Exploit **buffer overflow** in fingerd
  - Exploit **trapdoor** in sendmail
- Flaws in fingerd and sendmail were well-known, but not widely patched

Bootstrap Loader

- Once Morris worm got access…
- "Bootstrap loader" sent to victim
  - 99 lines of C code
- Victim compiled and executed code
- Bootstrap loader fetched the worm
- Victim **authenticated** sender
  - Don't want user to get a bad worm…

How to Remain Undetected?

- If transmission interrupted, all code deleted
- Code encrypted when downloaded
- Code deleted after decrypt/compile
- When running, worm regularly changed name and process identifier (PID)

Morris Worm: Bottom Line

- Shock to the Internet community of 1988
  - Internet of 1988 ***much*** different than today
- Internet designed to survive nuclear war
  - Yet, brought down by one graduate student!
  - At the time, Morris' father worked at NSA…
- Could have been much worse
- Result? CERT, more security awareness
- But should have been a wakeup call

Code Red Worm

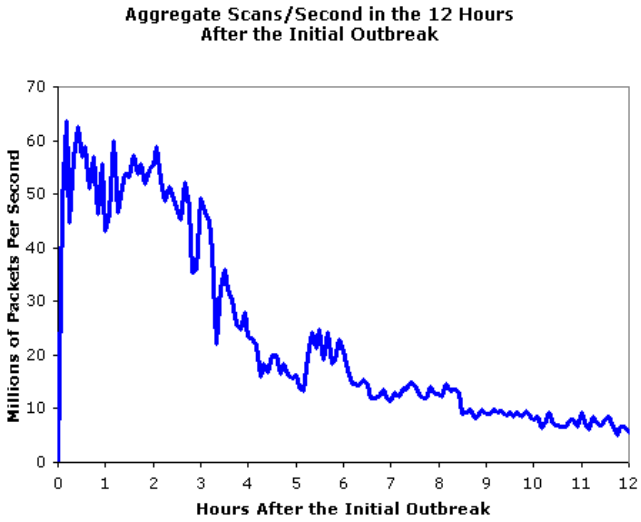
- Appeared in July 2001
- Infected more than **250,000 systems in about 15 hours**
- Eventually infected 750,000 out of about 6,000,000 vulnerable systems
- Exploited buffer overflow in Microsoft IIS server software
  - Then monitor traffic on port 80, looking for other susceptible servers

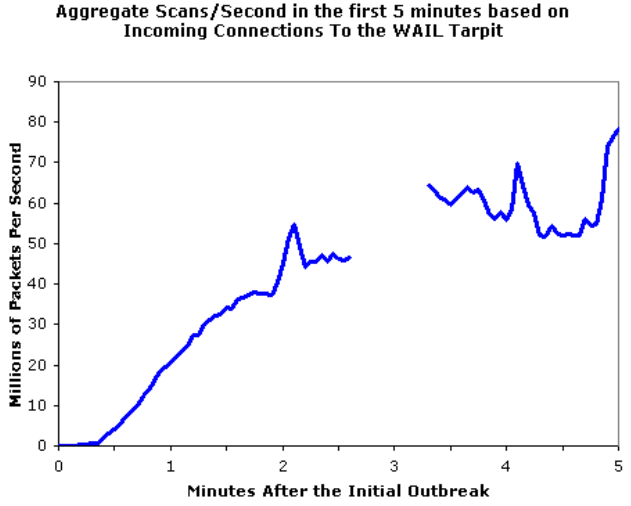
Code Red: What it Did

- Day 1 to 19 of month: spread its infection
- Day 20 to 27: distributed denial of service attack (DDoS) on www.whitehouse.gov
- Later version (several variants)
  - Included trapdoor for remote access
  - Rebooted to flush worm, leaving only trapdoor
- Some said it was "beta test for info warfare"
  - But, no evidence to support this

SQL Slammer

- Infected **75,000 systems in 10 minutes!**
- At its peak, infections doubled every 8.5 seconds
- Spread "too fast"…
- …so it "burned out" available bandwidth





Why was Slammer Successful?

- Worm size: **one 376-byte UDP packet**
- Firewalls often let one packet thru
  - Then monitor ongoing "connections"
- Expectation was that much more data required for an attack
  - So no need to worry about 1 small packet
- Slammer defied "experts"

Stuxnet

- Malware for information warfare…
- Discovered in 2010
  - Origins go back to 2008, or earlier
- Apparently, targeted Iranian nuclear processing facility
  - Reprogrammed specific type of PLC
  - Changed speed of centrifuges, causing damage to about 1000 of them

Stuxnet

- Many advanced features including…
  - Infect system via removable drives — able to get behind "airgap" firewalls
  - Used 4 unpatched MS vulnerabilities
  - Updates via P2P over a LAN
  - Contact C&C server for code/updates
  - Includes a Windows rootkit for stealth
  - Significant exfiltration/recon capability
  - Used a compromised private key

Malware Related to Stuxnet

- Duqu (2011)
  - Likely that developers had access to Stuxnet source code
  - Apparently, used mostly for info stealing
- Flame (2012)
  - May be “most complex” malware ever
  - Very sophisticated spyware mechanisms

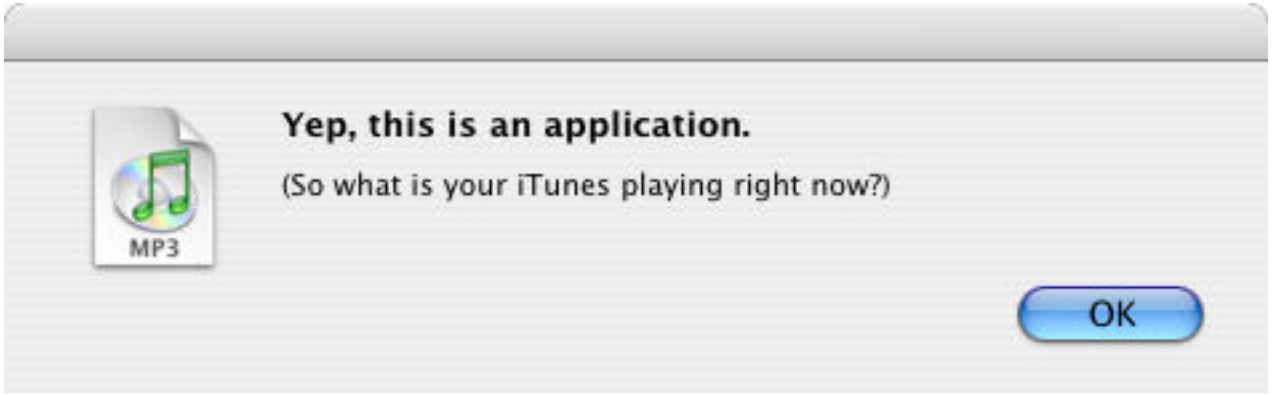
Trojan Horse Example

- Trojan: unexpected functionality
- Prototype trojan for the Mac
- File icon for freeMusic.mp3:

- For a real mp3, double click on icon
  - iTunes opens
  - Music in mp3 file plays
- But for freeMusic.mp3, unexpected results…

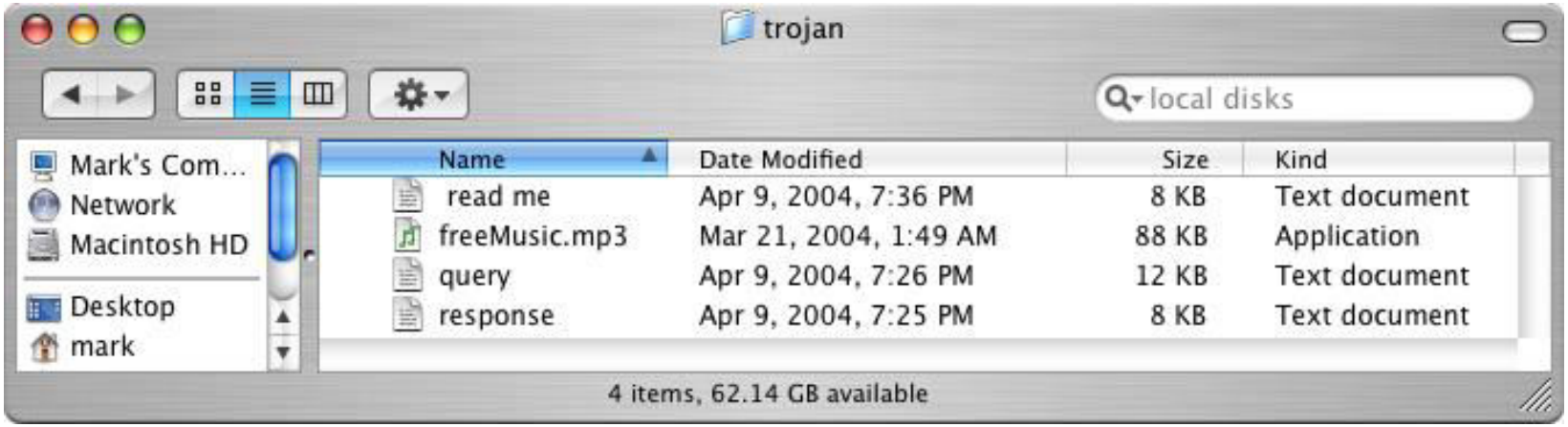
Mac Trojan

- Double click on freeMusic.mp3
  - iTunes opens (expected)
  - "Wild Laugh" (not expected)
  - Message box (not expected)



Trojan Example

- How does freeMusic.mp3 trojan work?
- This "mp3" is an application, not data



- This trojan is harmless, but…
- …could have done anything user could do
  - Delete files, download files, launch apps, etc.

Malware Detection

- Three common detection methods
  - Signature detection
  - Change detection
  - Anomaly detection
- We briefly discuss each of these
  - And consider advantages…
  - …and disadvantages

Signature Detection

- A **signature** may be a string of bits in exe
  - Might also use wildcards, hash values, etc.
- For example, W32/Beast virus has signature
  83EB 0274 EB0E 740A 81EB 0301 0000
  - That is, this string of bits appears in virus
- We can search for this signature in all files
- If string found, have we found W32/Beast?
  - Not necessarily — string could be in normal code
  - At random, chance is only $1/2^{112}$
  - But software is not random…

Signature Detection

- Advantages
  - Effective on "ordinary" malware
  - Minimal burden for users/administrators
- Disadvantages
  - Signature file can be large (10s of thousands)…
  - …making scanning slow
  - Signature files must be kept up to date
  - ***Cannot detect unknown viruses***
  - Cannot detect some advanced types of malware
- The most popular detection method

Change Detection

- Viruses must live somewhere
- If you detect a file has changed, it might have been infected
- How to detect changes?
  - Hash files and (securely) store hash values
  - Periodically re-compute hashes and compare
  - If hash changes, file **might** be infected

Change Detection

- Advantages
  - Virtually no false negatives
  - Can even detect previously unknown malware
- Disadvantages
  - Many files change — and often
  - Many false alarms (false positives)
  - Heavy burden on users/administrators
  - If suspicious change detected, then what? Might fall back on signature detection

Anomaly Detection

- Monitor system for anything "unusual" or "virus-like" or "potentially malicious" or …
- Examples of anomalous things
  - Files change in some unexpected way
  - System misbehaves in some way
  - Unexpected network activity
  - Unexpected file access, etc., etc., etc., etc.
- But, we must first define "normal"
  - And normal can (and must) change over time

Anomaly Detection

- Advantages
  - Chance of detecting unknown malware
- Disadvantages
  - No proven track record
  - Trudy can make abnormal look normal (go slow)
  - Must be combined with another method (e.g., signature detection)
- Also popular in intrusion detection (IDS)
- Difficult unsolved (unsolvable?) problem
  - Reminds me of AI…

Next…Future of Malware

- Recent trends
  - Encrypted, polymorphic, metamorphic malware
  - Fast replication/Warhol worms
  - Flash worms, slow worms
  - Botnets
- The future is bright for malware
  - Good news for the bad guys…
  - …bad news for the good guys
- Future of malware detection?