

Why Software?

- ❑ Why is software as important to security as crypto, access control, protocols?
- ❑ Virtually all information security features are implemented in software
- ❑ If your software is subject to attack, your security can be broken
 - Regardless of strength of crypto, access control, or protocols
- ❑ Software is a poor **foundation** for security

Bad Software is Ubiquitous

- ❑ NASA Mars Lander (cost \$165 million)
 - Crashed into Mars due to...
 - ...error in converting English and metric units of measure
- ❑ Denver airport
 - Baggage handling system — very buggy software
 - Delayed airport opening by 11 months
 - Cost of delay exceeded \$1 million/day
 - What happened to person responsible for this fiasco?
- ❑ Patriot Missiles
 - Gulf War 1991, used to intercept and destroy enemy missiles
 - 28 people died for a missed intercept
 - Count time using just 24 bits
 - They count the time in $1/10^{\text{th}}$ seconds...

Software Issues

Alice and Bob

- ❑ Find bugs and flaws by accident
- ❑ Hate bad software...
- ❑ ...but they learn to live with it
- ❑ Must make bad software work

Trudy

- ❑ Actively looks for bugs and flaws
- ❑ Likes bad software...
- ❑ ...and tries to make it misbehave
- ❑ Attacks systems via bad software

Complexity

- "Complexity is the enemy of security", Paul Kocher, Cryptography Research, Inc.

System	Lines of Code (LOC)
Assembly "Hello World"	10
Linux v1	10,000
iPhone App (average)	50,000
Photoshop v1	100,000
Space Shuttle	400,000
Windows 3.1	2.5 million
HD DVD player	4.5 million

A million of lines of code is like a book with 18,000 pages

Complexity

- "Complexity is the enemy of security", Paul Kocher, Cryptography Research, Inc.

System	Lines of Code (LOC)
Healthcare.gov	5 million
Google Chrome	7 million
Android	12 million
Boeing 787	14 million
MySQL	13 million
Linux kernel 3.1	15 million

Complexity

- "Complexity is the enemy of security", Paul Kocher, Cryptography Research, Inc.

System	Lines of Code (LOC)
Windows XP / 7	40 million
Microsoft Office	45 million
Microsoft Visual Studio	50 million
Facebook	61 million
Mac OS X Tiger	85 million
Car Software	100 million
Google	2 billion

Lines of Code and Bugs

- ❑ Conservative estimate: 5 bugs/10,000 LOC
- ❑ **Do the math**
 - Typical computer: 3k exe's of 100k LOC each
 - Conservative estimate: 50 bugs/exe
 - Implies about 150k bugs per computer
 - So, 30,000-node network has 4.5 billion bugs
 - Maybe only 10% of bugs security-critical and only 10% of those remotely exploitable
 - Then "only" 45 million critical security flaws!

Software Security Topics

- ❑ Program flaws (unintentional)
 - Buffer overflow
 - Incomplete mediation
 - Race conditions
- ❑ Malicious software (intentional)
 - Viruses
 - Worms
 - Other breeds of malware

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

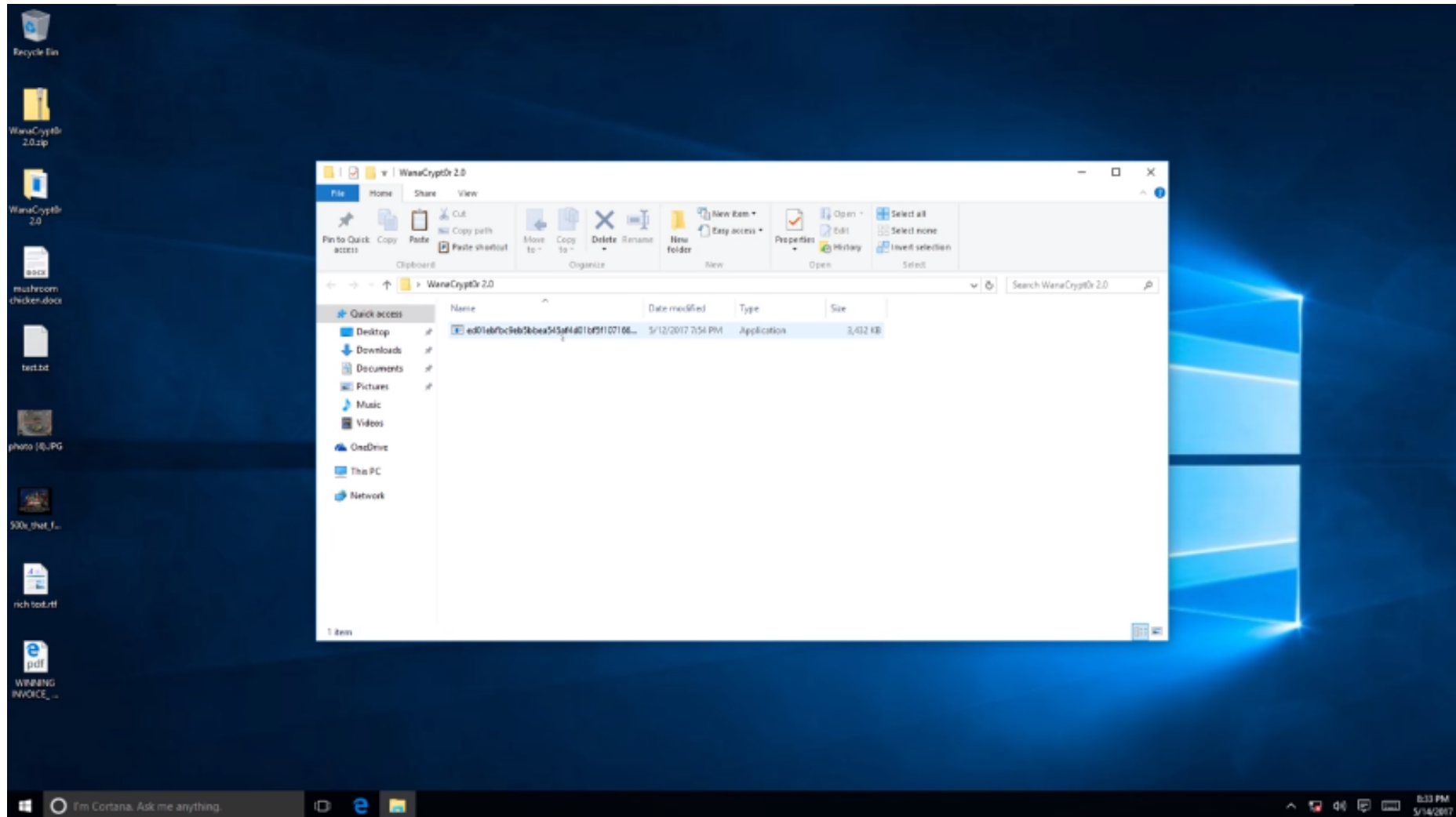
Where do Viruses Live?

- ❑ 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!

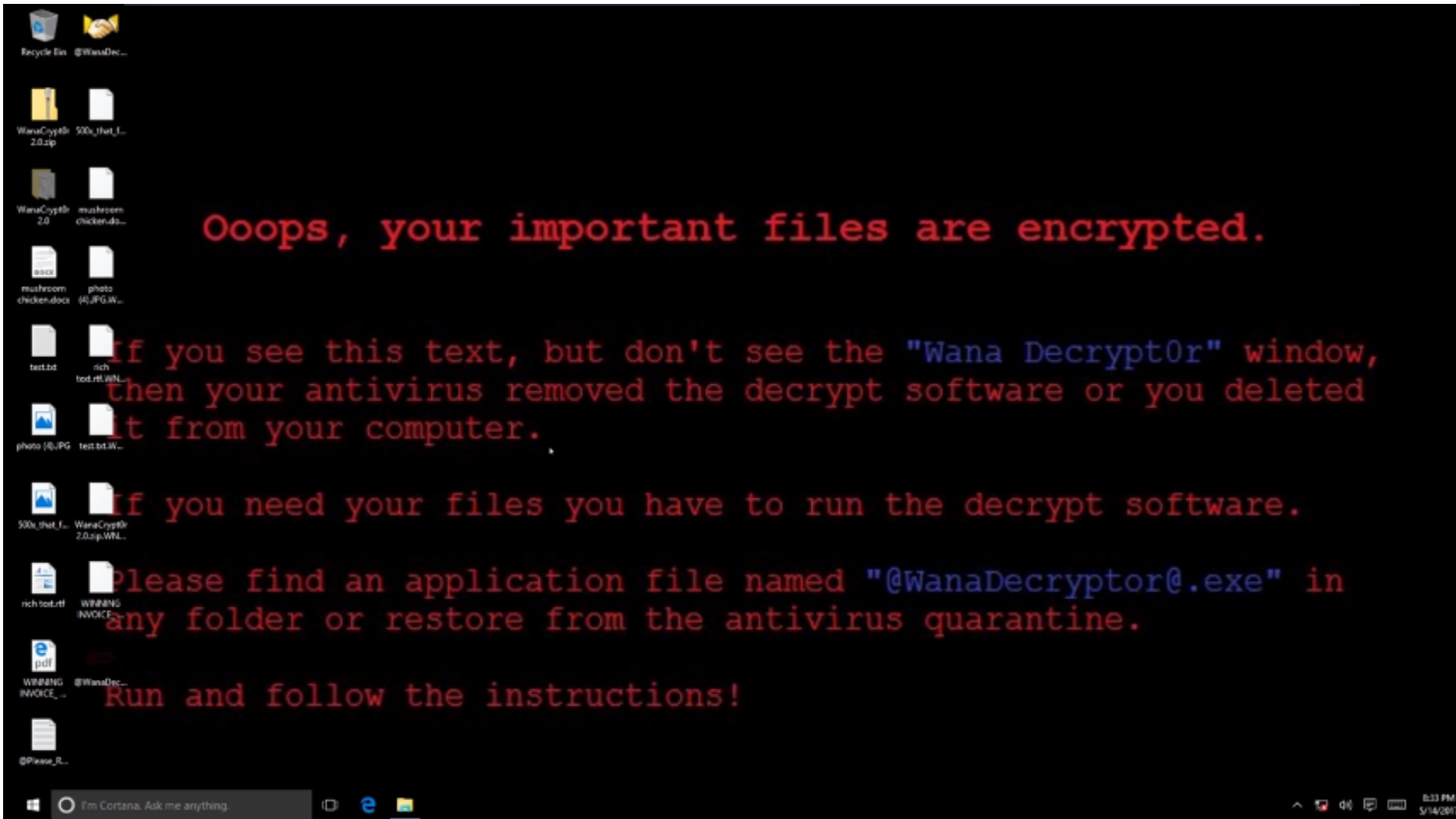
Ransomware



❑ Trojan.Ransom.WannaCrypt



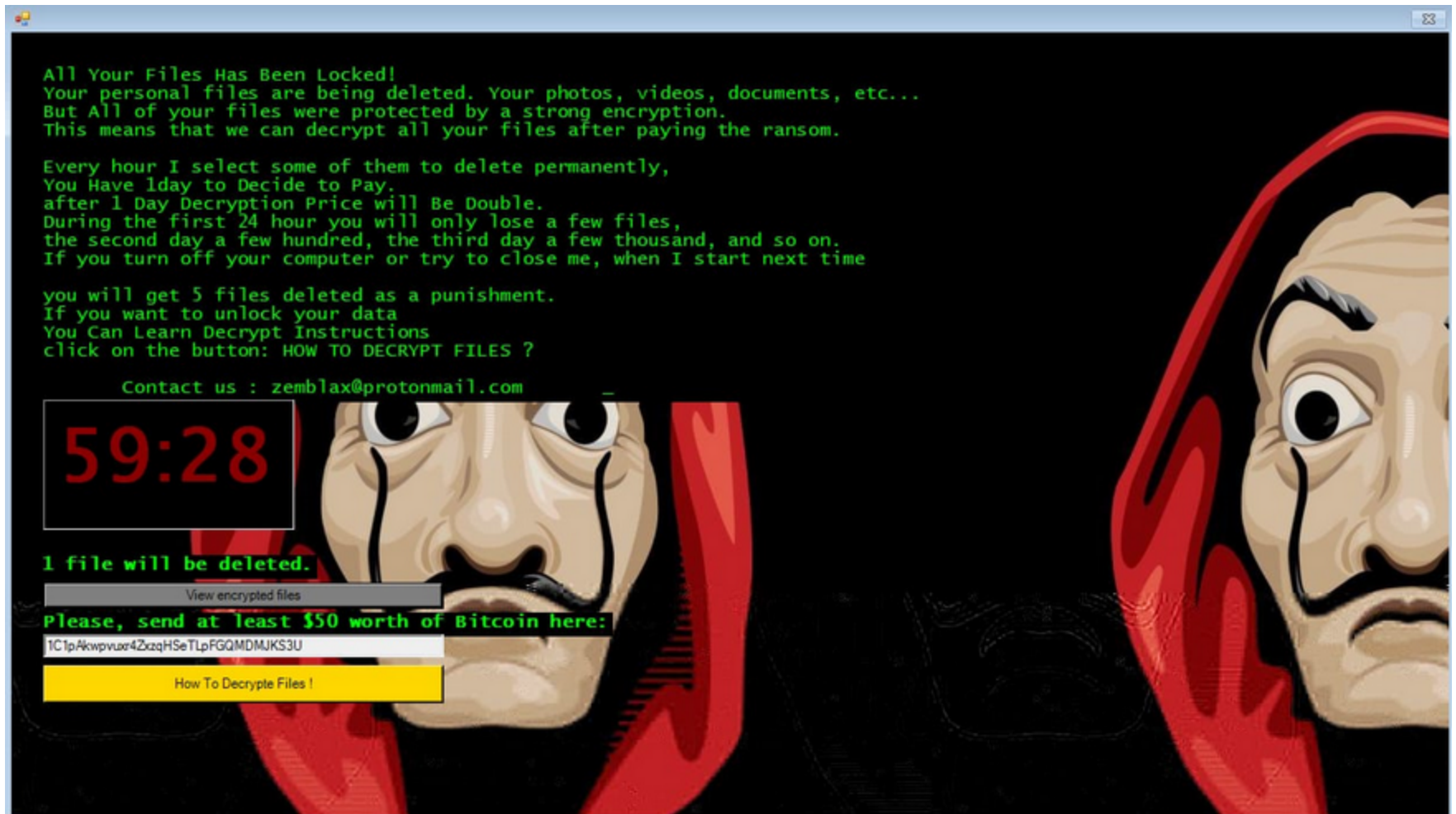
Trojan.Ransom.WannaCrypt



❑ Trojan.Ransom.WannaCrypt



Lokibot/Jigsaw (May 2020)



Ragnar Locker Ransomware attack on Capcom (November 2020)

- ❑ The list of potentially exposed records:
 - Japan's customer service video game support, help desk: 134,000 items, including names, addresses, phone numbers, email addresses
 - North America: Capcom Store member information: 14,000 items, including names, dates of birth, email addresses
 - Esports operations website members: 4,000 items, including names, email addresses, gender
 - Shareholder lists: 40,000 items, including names, addresses, shareholder numbers, amounts
 - Former employees and family: 28,000 people, applicant data (125,000 people): names, dates of birth, addresses, phone numbers, and more
 - Human resources data: 14,000 people
 - Confidential corporate information: business partner records, sales documents, and more
- ❑ Capcom is keen to emphasize that no credit card data has been included in the breach, as payments are managed by a third-party.

Ragnar Locker Ransomware attack on Capcom (November 2020)

- ❑ Monster Hunter Rise is coming to PC in October
- ❑ Resident Evil Village is planned for an April 2021 launch
- ❑ Monster Hunter Stories 2 is also slated for June 2021 on Nintendo Switch, and it will launch on PC as well

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

Brain

```
Microsoft(R) MS-DOS(R) Version 3.30  
(C)Copyright Microsoft Corp 1981-1987
```

```
A>dir/w
```

```
Volume in drive A is (C) BRAIN  
Directory of A:\
```

4201	CPI	5202	CPI	ANSI	SYS	APPEND	EXE	ASSIGN	COM
ATTRIB	EXE	CHKDSK	COM	COMMAND	COM	COMP	COM	COUNTRY	SYS
DISKCOMP	COM	DISKCOPY	COM	DISPLAY	SYS	DRIVER	SYS	EDLIN	COM
EXE2BIN	EXE	FASTOPEN	EXE	FDISK	COM	FIND	EXE	FORMAT	COM
GRAFTABL	COM	GRAPHICS	COM	JOIN	EXE	KEYB	COM	LABEL	COM
MODE	COM	MORE	COM	NLSFUNC	EXE	PRINT	COM	RECOVER	COM
SELECT	COM	SORT	EXE	SUBST	EXE	SYS	COM		

```
34 File(s)      5120 bytes free
```

```
A>_
```

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

Morris Worm built-in dictionary

aaa academia aerobics airplane albany albatross albert alex alexander algebra aliases alphabet ama amorphous analog anchor andromache animals answer anthropogenic anvils anything aria ariadne arrow arthur athena atmosphere aztecs azure bacchus bailey banana bananas bandit banks barber baritone bass bassoon batman beater beauty beethoven beloved benz beowulf berkeley berliner beryl beverly bicameral bob brenda brian bridget broadway bumbling burgess campanile cantor cardinal carmen carolina caroline cascades castle cat cayuga celtics cerulean change charles charming charon chester cigar classic clusters coffee coke collins comrades **computer** condo cookie cooper cornelius couscous creation creosote cretin daemon dancer daniel danny dave december defoe deluge desperate develop dieter digital discovery disney dog drought duncan eager easier edges edinburgh edwin edwina egghead eiderdown eileen einstein elephant elizabeth ellen emerald engine engineer enterprise enzyme ersatz establish estate euclid evelyn extension fairway felicia fender fermat fidelity finite fishers flakes float flower flowers foolproof football foresight format forsythe fourier fred friend frighten fun fungible gabriel gardner garfield gauss george gertrude ginger glacier gnu golfer gorgeous gorges gosling gouge graham gryphon guest guitar gumption guntis hacker hamlet handily happening harmony harold harvey hebrides heinlein hello help herbert hiawatha hibernia honey horse horus hutchins imbroglio imperial include ingres inna innocuous irishman isis japan jessica jester jixian johnny joseph joshua judith juggle julia kathleen kermit kernel kirkland knight ladle lambda lamination larkin larry lazarus lebesgue lee leland leroy lewis light lisa louis lynne macintosh mack maggot magic malcolm mark markus marty marvin master maurice mellon merlin mets michael michelle mike minimum minsky moguls moose morley mozart nancy napoleon nepenthe ness network newton next noxious nutrition nyquist oceanography ocelot olivetti olivia oracle orca orwell osiris outlaw oxford pacific painless pakistan pam papers **password** patricia penguin peoria percolate persimmon persona pete peter philip phoenix pierre **pizza** plover plymouth polynomial pondering pork poster praise precious prelude prince princeton protect protozoa pumpkin puneet puppet rabbit rachmaninoff rainbow raindrop raleigh random rascal really rebecca remote rick ripple robotics rochester rolex romano ronald rosebud rosemary roses ruben rules ruth sal saxon scamper scheme scott scotty **secret** sensor serenity sharks sharon sheffield sheldon shiva shivers shuttle signature simon simple singer single smile smiles smooch smother snatch snoopy soap socrates sossina sparrows spit spring springer squires strangle stratford stuttgart subway success summer super superstage support supported surfer suzanne swearer symmetry tangerine tape target tarragon taylor telephone temptation thailand tiger toggle tomato topography tortoise toyota trails trivial trombone tubas tuttle umesh unhappy unicorn unknown urchin utility vasant vertigo vicky village virginia warren water weenie whatnot whiting whitney will william williamsburg willie winston wisconsin wizard wombat woodwind wormwood yacov yang yellowstone yosemite zap zimmerman

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

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 (Computer Emergency Response Team), more security awareness
- ❑ But should have been a wakeup call

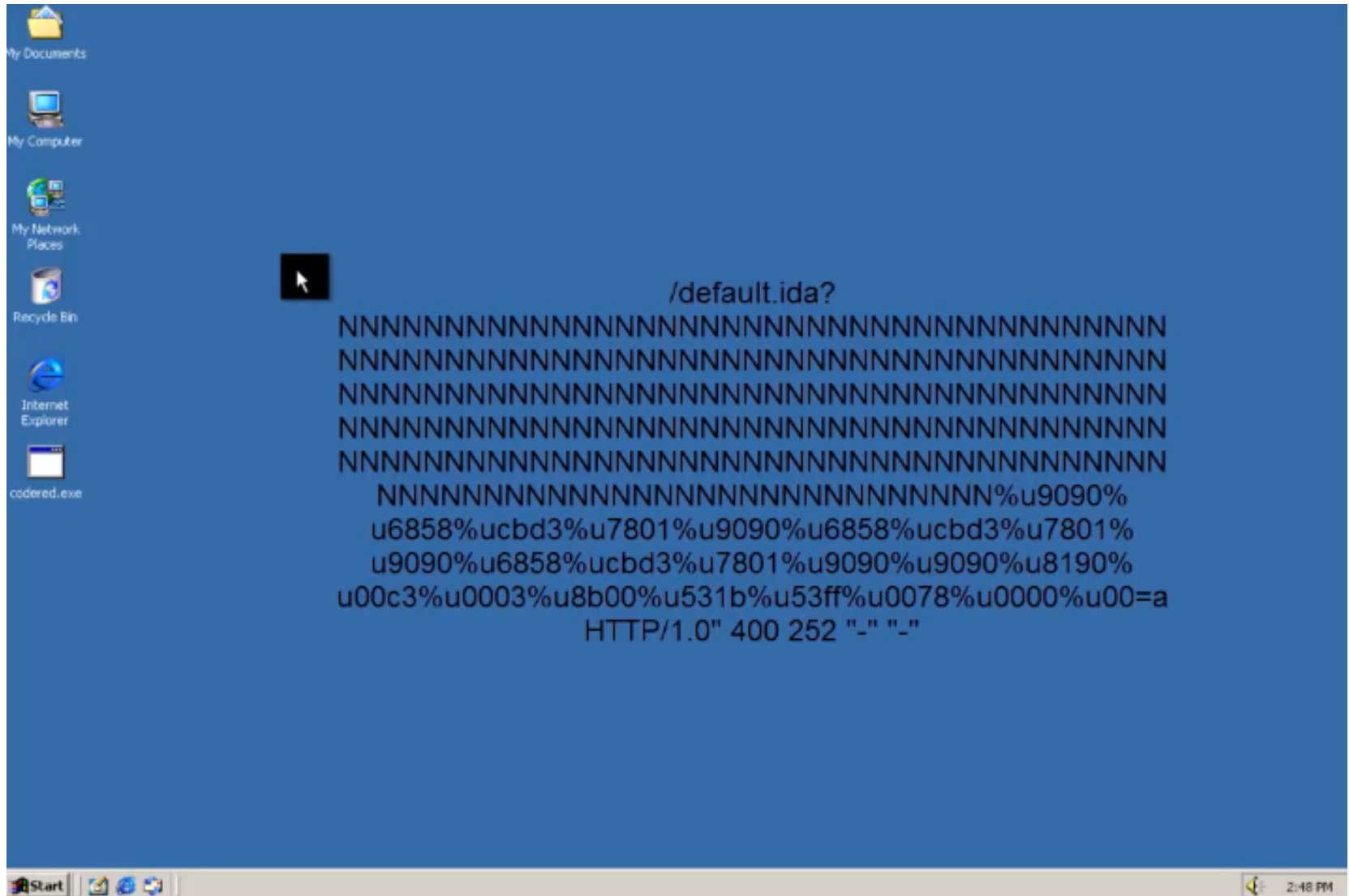
Code Red Worm

- ❑ 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

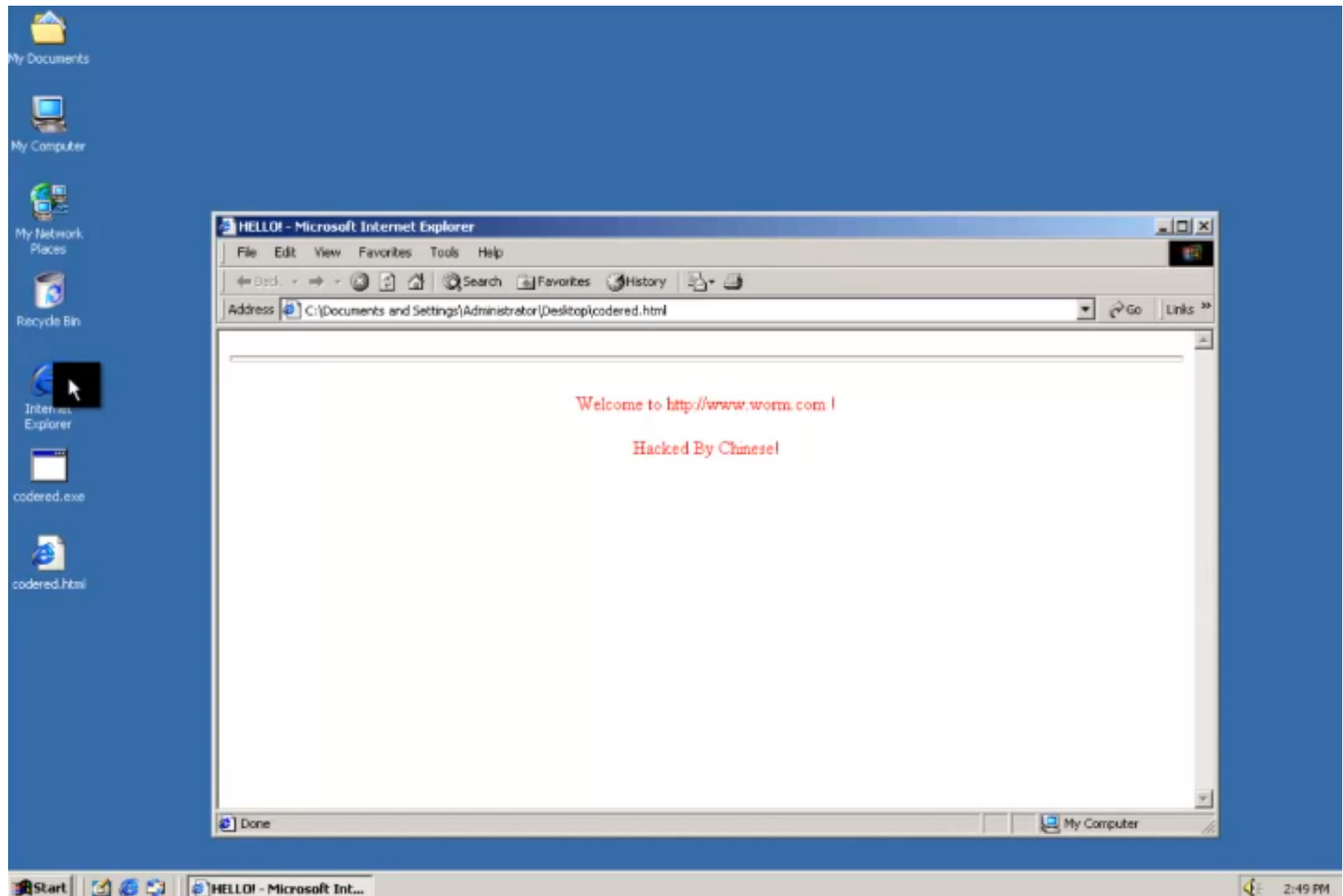
Code Red: What it Did

- ❑ 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

Code Red



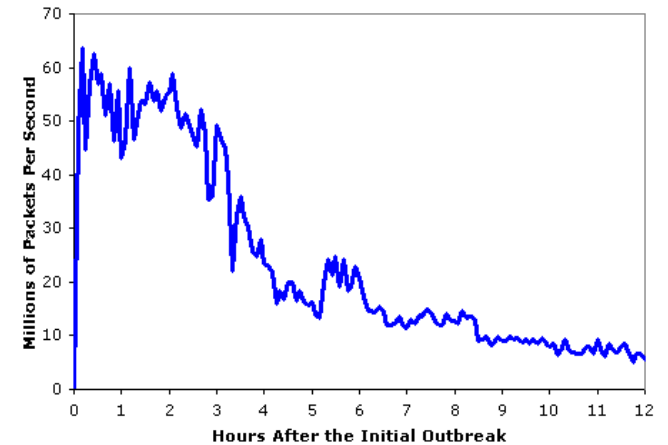
Code Red



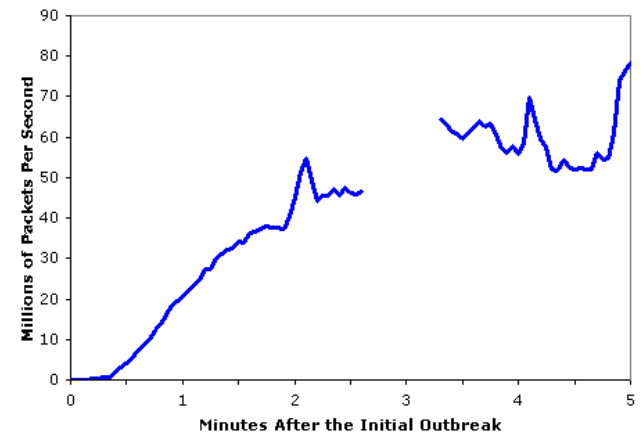
SQL Slammer

- ❑ 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

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



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)

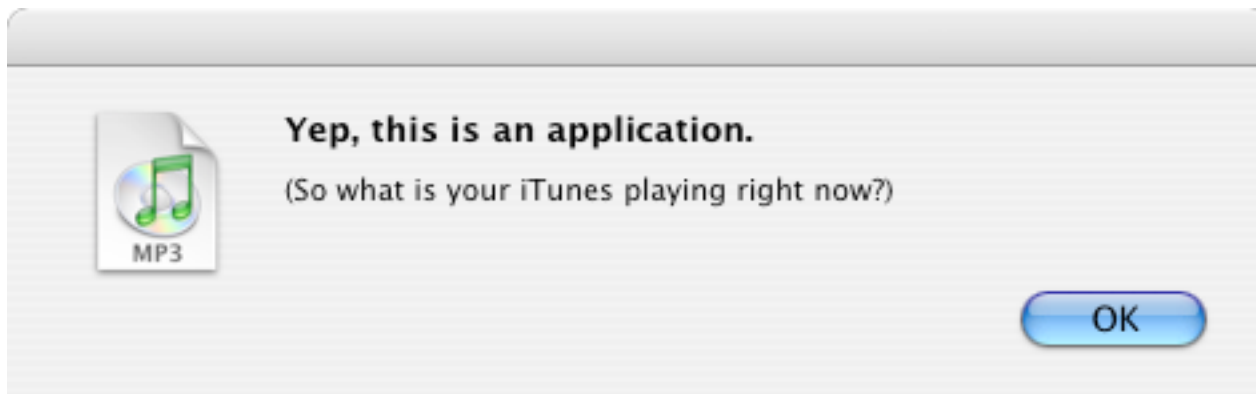
- May be “most complex” malware ever
- Very sophisticated spyware mechanisms

Trojan Horse Example

- ❑ Trojan: unexpected functionality
- ❑ Prototype trojan for the Mac
- ❑ File icon for freeMusic.mp3:
freeMusic.mp3
- ❑ For a real mp3, double click on icon
 - iTunes opens
 - Music in mp3 file plays
- ❑ But for freeMusic.mp3, unexpected results...

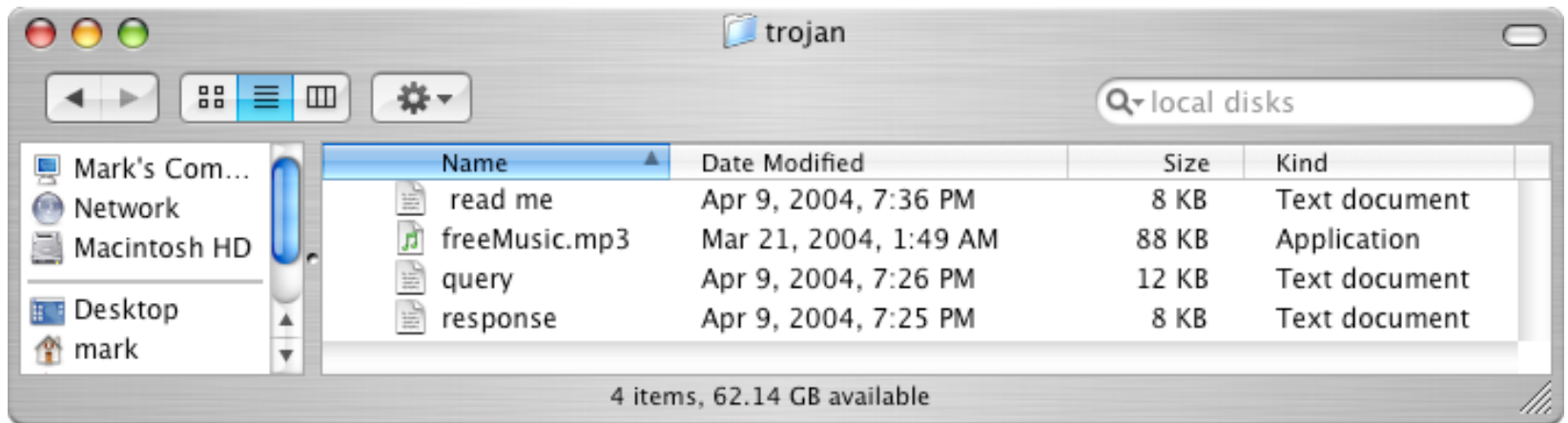
Mac Trojan

- ❑ 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

Encrypted Viruses

- ❑ Virus writers know **signature detection** used
- ❑ So, how to evade signature detection?
- ❑ Encrypting the virus is a good approach
 - Ciphertext looks like random bits
 - Different key, then different "random" bits
 - So, different copies have no common signature
- ❑ Encryption often used in viruses today

Encrypted Viruses

- ❑ How to detect encrypted viruses?
- ❑ Scan for the decryptor code
 - More-or-less standard signature detection
 - But may be more false alarms
- ❑ Why not encrypt the decryptor code?
 - Then encrypt the decryptor of the decryptor (and so on...)
- ❑ Encryption of limited value to virus writers

Polymorphic Malware

❑ Polymorphic worm

- Body of worm is encrypted
- Decryptor code is “mutated” (or “morphed”)
- Trying to hide decryptor signature
- Like an encrypted worm on steroids...

Q: How to detect?

A: Emulation — let the code decrypt itself

- Slow, and anti-emulation is possible

Metamorphic Malware

- ❑ A metamorphic worm mutates before infecting a new system
 - Sometimes called “body polymorphic”
- ❑ Such a worm can, in principle, evade signature-based detection
- ❑ Mutated worm must function the same
 - And be “different enough” to avoid detection
- ❑ Detection is a difficult research problem

Metamorphic Worm

- ❑ One approach to metamorphic replication...
 - The worm is disassembled
 - Worm then stripped to a base form
 - Random variations inserted into code (permute the code, insert dead code, etc., etc.)
 - Assemble the resulting code
- ❑ Result is a worm with same functionality as original, but different signature

Miscellaneous Attacks

- ❑ Numerous attacks involve software
- ❑ We'll discuss a few issues that do not fit into previous categories
 - Salami attack
 - Linearization attack
 - Time bomb

Salami Attack

- ❑ What is Salami attack?
 - Programmer “slices off” small amounts of money
 - Slices are hard for victim to detect
- ❑ Example
 - Bank calculates interest on accounts
 - Programmer “slices off” any fraction of a cent and puts it in his own account
 - No customer notices missing partial cent
 - Bank may not notice any problem
 - Over time, programmer makes lots of money!

Salami Attack

- ❑ Such attacks are possible for insiders
- ❑ Do salami attacks actually occur?
 - Or is it just Office Space folklore?
- ❑ Programmer added a few cents to every employee payroll tax withholding
 - But money credited to programmer's tax
 - Programmer got a big tax refund!
- ❑ Rent-a-car franchise in Florida inflated gas tank capacity to overcharge customers

Salami Attacks

- ❑ Employee reprogrammed Taco Bell cash register: \$2.99 item registered as \$0.01
 - Employee pocketed \$2.98 on each such item
 - A large “slice” of salami!
- ❑ In LA, four men installed computer chip that overstated amount of gas pumped
 - Customers complained when they had to pay for more gas than tank could hold
 - Hard to detect since chip programmed to give correct amount when 5 or 10 gallons purchased
 - Inspector usually asked for 5 or 10 gallons

Linearization Attack

- ❑ Program checks for serial number S123N456
- ❑ For efficiency, check made one character at a time
- ❑ Can attacker take advantage of this?

```
#include <stdio.h>

int main(int argc, const char *argv[])
{
    int i;
    char serial[9]="S123N456\n";

    for(i = 0; i < 8; ++i)
    {
        if(argv[1][i] != serial[i]) break;
    }
    if(i == 8)
    {
        printf("\nSerial number is correct!\n\n");
    }
}
```

Linearization Attack

- ❑ Correct number takes longer than incorrect
- ❑ Trudy tries all 1st characters
 - Find that S takes longest
- ❑ Then she guesses all 2nd characters: S*
 - Finds S1 takes longest
- ❑ And so on...
- ❑ Trudy can recover one character at a time!
 - Same principle as used in lock picking

Linearization Attack

- ❑ What is the advantage to attacking serial number one character at a time?
- ❑ Suppose serial number is 8 characters and each has 128 possible values
 - Then $128^8 = 2^{56}$ possible serial numbers
 - Attacker would guess the serial number in about 2^{55} tries — a lot of work!
 - Using the linearization attack, the work is about $8 * (128/2) = 2^9$ which is easy

Linearization Attack

- ❑ A real-world linearization attack
- ❑ TENEX (an ancient timeshare system)
 - Passwords checked one character at a time
 - Careful timing was **not** necessary, instead...
 - ...could arrange for a “page fault” when next unknown character guessed correctly
 - Page fault register was user accessible
- ❑ Attack was very easy in practice

Time Bomb

- ❑ In 1986 Donald Gene Burleson told employer (USPA and IRA Company) to stop withholding taxes from his paycheck
- ❑ His company refused
- ❑ He planned to sue his company
 - He used company time to prepare legal docs
 - Company found out and fired him
- ❑ Burleson had been working on malware...
 - After being fired, his software "time bomb" deleted 168,000 payroll records

Time Bomb

- ❑ Company was reluctant to pursue the case
- ❑ So Burleson sued company for back pay!
 - Then company finally sued Burleson
- ❑ In 1988 Burleson fined \$11,800
 - Case took years to prosecute...
 - Cost company thousands of dollars...
 - Resulted in a slap on the wrist for attacker
- ❑ One of the first computer crime cases
- ❑ Many cases since follow a similar pattern
 - Companies reluctant to prosecute