

# CSE 3320

## *Chapter 16: Protection and Security*

Trevor Bakker

The University of Texas at Arlington

The three golden rules to ensure computer security are:

- do not own a computer;
- do not power it on;
- and do not use it.

Robert Morris, Chief Scientist  
National Security Agency's  
National Computer Security Center

# Why shouldn't we blindly trust all programs?

Programmers can be bored, exhausted, lazy, careless, ignorant, unintelligent, malicious, or thoroughly evil.

Any of these can produce work that can damage our work or our entire system.

# Who we should watch out for

- Generally, there are two groups of people we need to worry about:
- Hackers - Dangerous for home users. They attack system weaknesses that most home users are not knowledgeable enough to recognize, much less fix.
  - Most notorious
  - Only a portion of the problem

# Who we should watch out for

- Legitimate users attempting unauthorized use.
  - Sending abusive email
  - Stealing money
  - Wasting time surfing
  - Playing games
  - Snooping on personal information
  - Stealing national secrets



# Exploits

- Hackers are generally exploiting problems in the OS.
- Executing operations that they are not supposed to be able to execute.
- Through exploits can gain access to administrator rights

# Exploits

- OS vendors are usually quick to patch
  - Some users are slow to patch
    - Fixes are not as well tested as a full release
    - Not willing to take a risk with an untested patch on a large install base.
  - Lazy

# Malware

- General category of exploit
  - Virus
  - Worms
  - Trojans
  - Spyware



# Virus

- **Computer virus** are portions of programs that insert themselves into other programs in a manner analogous to biological viruses.
- When a program containing a virus is run on a computer the virus will insert itself into other programs on secondary storage.
- Hardest part for the virus writer is to getting the user to run the program containing the virus

# Virus

- When floppy disks were common the most common place for a virus was the boot sector
- A [memory-resident virus](#) installs itself as part of the operating system when executed, after which it remains in RAM from the time the computer is booted up to when it is shut down.
- Resident viruses overwrite interrupt handling code or other functions, and when the operating system attempts to access the target file or disk sector, the virus code intercepts the request and redirects the control flow to the replication module, infecting the target.
- Non-memory-resident virus when executed, scans the disk for targets, infects them, and then exits. It does not remain in memory after it is done executing.

# Virus

- Viruses often perform some type of harmful activity on infected hosts, such as stealing hard disk space or CPU time, accessing private information, corrupting data, displaying political or humorous messages on the user's screen, spamming their contacts, or logging their keystrokes.
- Not all viruses carry a destructive payload or attempt to hide themselves—the defining characteristic of viruses is that they are self-replicating computer programs which install themselves without the user's consent.

# Virus classification

- By target
  - boot sector: *infect a master boot record*
  - file infector: *infects executable OS files*
  - macro virus: *infects files to be used by an app*
  - multipartite: infects multiple ways
- By concealment
  - encrypted virus: *encrypted; key stored in virus*
  - stealth virus: *hides itself (e.g., compression)*
  - polymorphic virus: *recreates with diff “signature”*
  - metamorphic virus: *recreates with diff signature and behavior*

# First Virus

- The first academic work on the theory of self-replicating computer programs was done in 1949 by John von Neumann who gave lectures at the University of Illinois about the "Theory and Organization of Complicated Automata".



# First Virus

- The work of von Neumann was later published as the "Theory of self-reproducing automata". In his essay von Neumann described how a computer program could be designed to reproduce itself. Von Neumann's design for a self-reproducing computer program is considered the world's first computer virus, and he is considered to be the theoretical father of computer virology.

# Creeper Virus

- The Creeper virus was first detected on ARPANET, the forerunner of the Internet, in the early 1970s. Creeper was an experimental self-replicating program written by Bob Thomas at BBN Technologies in 1971.
- Creeper used the ARPANET to infect DEC PDP-10 computers running the TENEX operating system. Creeper gained access via the ARPANET and copied itself to the remote system where the message, "I'm the creeper, catch me if you can!" was displayed.
- The Reaper program was created to delete Creeper

# Michelangelo Virus

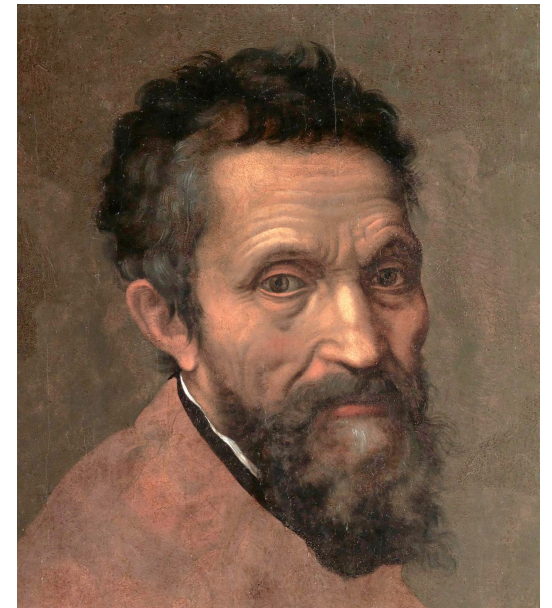
- First reported in April, 1991 in Sweden and the Netherlands.
- Michelangelo is a memory resident infector of diskette boot sectors and the hard disk master boot sector (partition table).
  - Roughly based on the Stoned virus, though very different in its behavior.





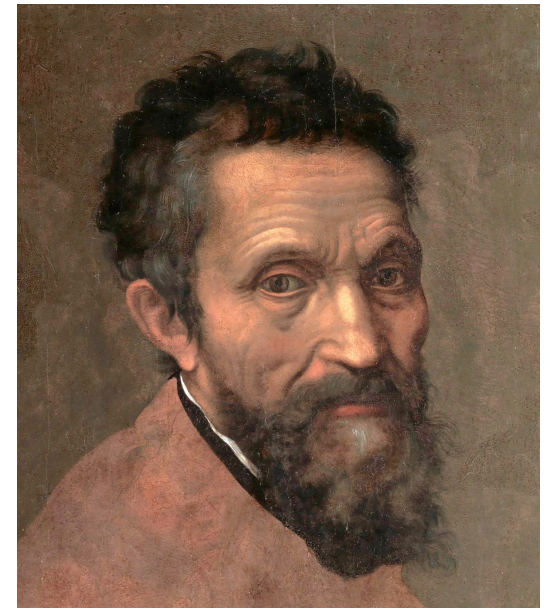
# Michelangelo Virus

- The Michelangelo virus becomes memory resident the first time the system is attempted to be booted with a Michelangelo infected disk.
- Regardless of whether this boot is successful, Michelangelo will become memory resident.



# Michelangelo Virus

- Once Michaelangelo is memory resident, it will infect diskette boot sectors as they are accessed.
- Master boot sector infections will result in the original master boot sector having been moved to Side 0, Cylinder 0, Sector 7 on the hard disk.
- Michelangelo activates on March 6, at which time it will format the system hard disk by overwriting it with random characters from system memory.



# Michelangelo Virus Source Code

# Melissa Virus

- In late March 1999, a programmer named David Lee Smith hijacked an America Online (AOL) account and used it to post a file on an Internet newsgroup named “alt.sex.”
- The posting promised dozens of free passwords to fee-based websites with adult content. When users took the bait, downloading the document and then opening it with Microsoft Word, a virus was unleashed on their computers.
- The Melissa virus started by taking over victims’ Microsoft Word program.
- It then used a macro to hijack their Microsoft Outlook email system and send messages to the first 50 addresses in their mailing lists.
- Those messages, in turn, tempted recipients to open a virus-laden attachment by giving it such names as “sexxy.jpg” or “naked wife” or by deceitfully asserting, “Here is the document you requested ... don’t show anyone else ;-).” With the help of some devious social engineering, the virus operated like a sinister, automated chain letter.

# Melissa Virus

- Email servers at more than 300 corporations and government agencies worldwide became overloaded.
- Approximately one million email accounts were disrupted, and Internet traffic in some locations slowed to a crawl.
- Within a few days, cybersecurity experts had mostly contained the spread of the virus and restored the functionality of their networks, although it took some time to remove the infections entirely.
- An estimated **\$80 million** for the cleanup and repair of affected computer systems.

Melissa Virus Code

# Worms

- A **worm** is a standalone malware computer program that replicates itself in order to spread to other computers.
- Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.
- Unlike a computer virus, it does not need to attach itself to an existing program.
- Worms almost always cause at least some harm to the network, even if only by consuming bandwidth.

# Worms

```
0 00 00-6D 73 62 6C msbl
0 6A 75-73 74 20 77 ast.exe I just w
9 20 4C-4F 56 45 20 ant to say LOVE
0 62 69-6C 6C 79 20 YOU SAN!! billy
0 64 6F-20 79 6F 75 gates why do you
3 20 70-6F 73 73 69 make this possi
0 20 6D-61 6B 69 6E ble ? Stop makin
E 64 20-66 69 78 20 g money and fix
7 61 72-65 21 21 00 your software!!
0 00 00-7F 00 00 00 ♠ ♡▶ H △
0 00 00-01 00 01 00 ð_ð_ ☹ ☹ ☹
0 00 00-00 00 00 46 á☹ L F
C C9 11-9F E8 08 00 ♦ lêèù-▯◀f▯
0 00 03-10 00 00 00 +▶H'☹ ♠ ♡▶
3 00 00-01 00 04 00 p♥ ð ð♥ ☹ ♦
```

Hex dump of the Blaster worm, showing a message left for Microsoft CEO Bill Gates by the worm programmer



# Worm Propagation Model

$$\frac{dI(t)}{dt} = \beta I(t) S(t)$$

where

$I(t)$  = number of individuals infected as of time  $t$

$S(t)$  = number of susceptible individuals (susceptible to infection but not yet infected) at time  $t$

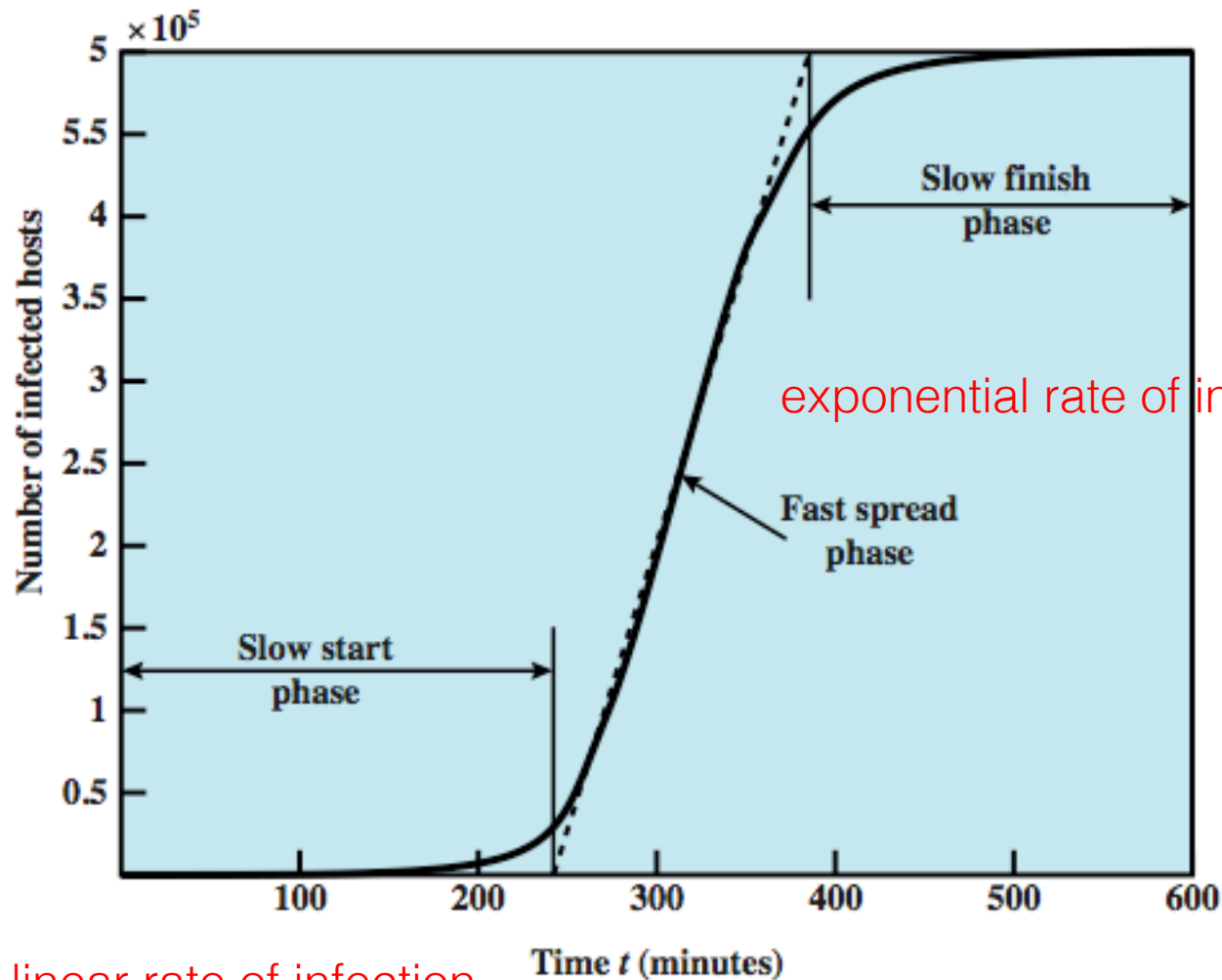
$\beta$  = infection rate

$N$  = size of the population,  $N = I(t) + S(t)$

Based on classic epidemic model

# Worm Propagation Model

(based on recent attacks)



linear rate of infection

# Morris Worm

- The Morris worm or Internet worm of November 2, 1988 was one of the first computer worms distributed via the Internet.
- It is considered the first worm and was certainly the first to gain significant mainstream media attention.
- It also resulted in the first conviction in the US under the 1986 Computer Fraud and Abuse Act.
- It was written by a student at Cornell University, Robert Tappan Morris, and launched on November 2, 1988 from MIT.

# Morris Work

- The Morris worm was not written to cause damage, but to gauge the size of the Internet. The worm was released from MIT to disguise the fact that the worm originally came from Cornell.
- It worked by exploiting known vulnerabilities in Unix sendmail, finger, and rsh/rexec, as well as weak passwords.

# Morris Worm

- The critical error that transformed the worm from a potentially harmless intellectual exercise into a virulent denial of service attack was in the spreading mechanism.
- The worm could have determined whether to invade a new computer by asking whether there was already a copy running. But just doing this would have made it trivially easy to kill: everyone could just run a process that would answer "yes" when asked whether there was already a copy, and the worm would stay away.

# Morris Worm

- To compensate for this possibility, Morris directed the worm to copy itself even if the response is "yes" 1 out of 7 times.
- This was way too excessive, and the worm spread rapidly, infecting some computers multiple times.
- Morris remarked, when he heard of the mistake, that he "should have tried it on a simulator first".

# Morris Worm

- The Internet was also partitioned for several days, as regional networks disconnected from the NSFNet backbone and from each other to prevent recontamination as they cleaned their own networks.
- The Morris worm prompted DARPA to fund the establishment of the CERT/CC at Carnegie Mellon University to give experts a central point for coordinating responses to network emergencies.
- Robert Morris was tried and convicted of violating United States Code: Title 18 (18 U.S.C. § 1030), the Computer Fraud and Abuse Act.
  - After appeals he was sentenced to three years probation, 400 hours of community service, a fine of \$10,050 plus the costs of his supervision.

# The Worm Author

Robert Morris is now  
a professor at MIT





# Trojans

- **Trojan**, in computing is a generally non-self-replicating type of malware program containing code that, when executed, carries out malicious actions.
- The term is derived from the story of the wooden horse used to trick defenders of Troy into taking concealed warriors into their city



# Trojans

- A Trojan often acts as a backdoor, contacting a controller which can then have unauthorized access to the affected computer.
- While Trojans are not easily detectable by themselves, computers may appear to run slower due to heavy processor or network usage.

# Spyware

- A special class of trojan.
- Relatively benign in the sense that they do not damage the computer they are running.
- Typically report local activity to a remote website

# Sony Rootkit

- The Sony BMG CD copy protection rootkit scandal of 2005–2007 was a deceptive, illegal, and harmful copy protection measure implemented by Sony BMG on about 22 million CDs.
- When inserted into a computer, the CDs installed one of two pieces of software which provided a form of digital rights management (DRM) by modifying the operating system to interfere with CD copying.

# Sony Rootkit

- Both programs could not be easily uninstalled, and they created vulnerabilities that were exploited by unrelated malware.
- Sony claims this was unintentional.
- One of the programs installed even if the user refused its EULA, and it "phoned home" with reports on the user's private listening habits;

# Sony Rootkit

- The other program was not mentioned in the EULA at all, contained code from several pieces of open-source software in an apparent infringement of copyright, and configured the operating system to hide the software's existence.

# Sony Rootkit

On a National Public Radio program, Thomas Hesse, President of Sony BMG's global digital business division asked, "Most people, I think, don't even know what a rootkit is, so why should they care about it?"

# Denial of Service

- **Denial-of-service** (DoS) or distributed denial-of-service (DDoS) attack is an attempt to make a machine or network resource unavailable to its intended users.
- Lots of different types of DoS. We'll talk about two.



# Ping of Death

- A correctly formed ping message is typically 84 bytes in size.
- Historically, many computer systems could not properly handle a ping packet larger than the maximum IPv4 packet size of 65535 bytes.
- Larger packets could crash the target computer.

# SYN Flood

- TCP connections start with a “three-way” handshake.
- The initiator sends a packets that contains a SYN flag which tells the receiver that a connection is forthcoming.
- The receiver allocates buffers and clears some data fields then sends a message to the originator.
- Once the originator receives this message it begins to send packets.

# SYN Flood

- In a SYN flood the originator never responds to the acknowledgement.
- As more and more SYN requests flood in the recipient either crashes or opens so many connections it can not service any additional requests.

# Buffer Overflows

- In order to do damage, a virus, worm, or hacker needs to fool the system into running code in supervisor mode.
- One of the most common ways is to exploit a coding error called a buffer overflow.
- A [buffer overflow](#) occurs when a process stores data beyond the bounds of a buffer.

# Buffer Overflow Example

```
char      A[8] = " ";
unsigned short B = 1979;
```

[illegible]

# Buffer Overflow Example

```
strcpy(A, "excessive");
```

| variable name | A   |     |     |     |     |     |     |     | B     |    |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-------|----|
| value         | 'e' | 'x' | 'c' | 'e' | 's' | 's' | 'i' | 'v' | 25856 |    |
| hex           | 65  | 78  | 63  | 65  | 73  | 73  | 69  | 76  | 65    | 00 |

By copying 10 bytes into an 8 byte array  
we've overwritten the value of B

# Buffer Overflow Code Examples

# Stack Buffer Overflow

- A **stack buffer overflow** or stack buffer overrun occurs when a program writes to a memory address on the program's call stack outside of the intended data structure; usually a fixed length buffer.
- Overfilling a buffer on the stack is more likely to derail program execution than overfilling a buffer on the heap because the stack contains the return addresses for all active function calls.
- Jackpot!

Overwrite the return  
address and you can  
execute any arbitrary code



# Stack Buffer Overflow Example

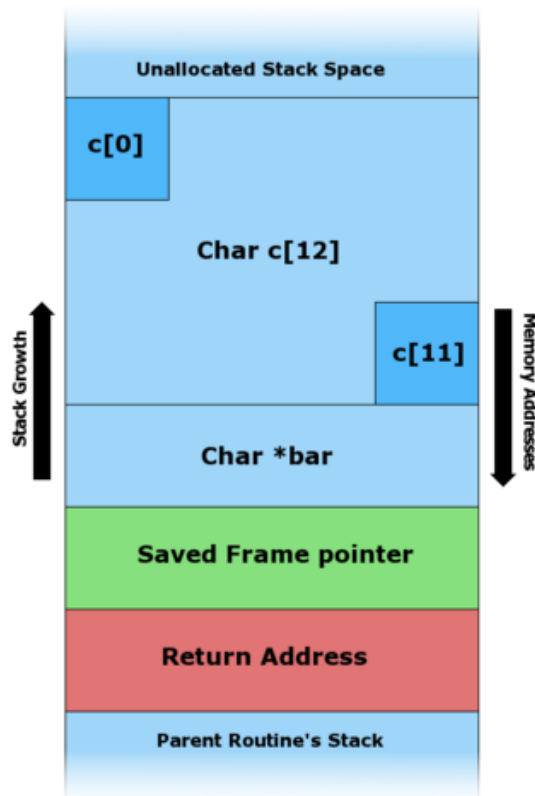
```
#include <string.h>

void foo (char *bar)
{
    char c[12];

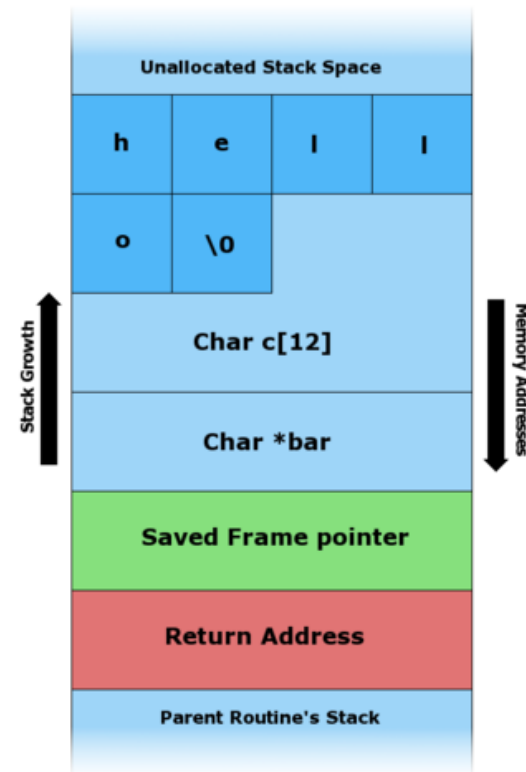
    strcpy(c, bar); // no bounds checking
}

int main (int argc, char **argv)
{
    foo(argv[1]);
}
```

# Stack Buffer Overflow Example

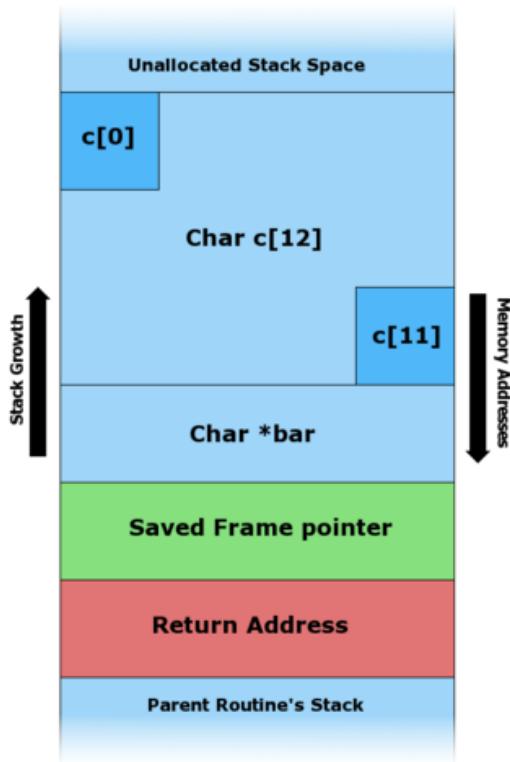


Before

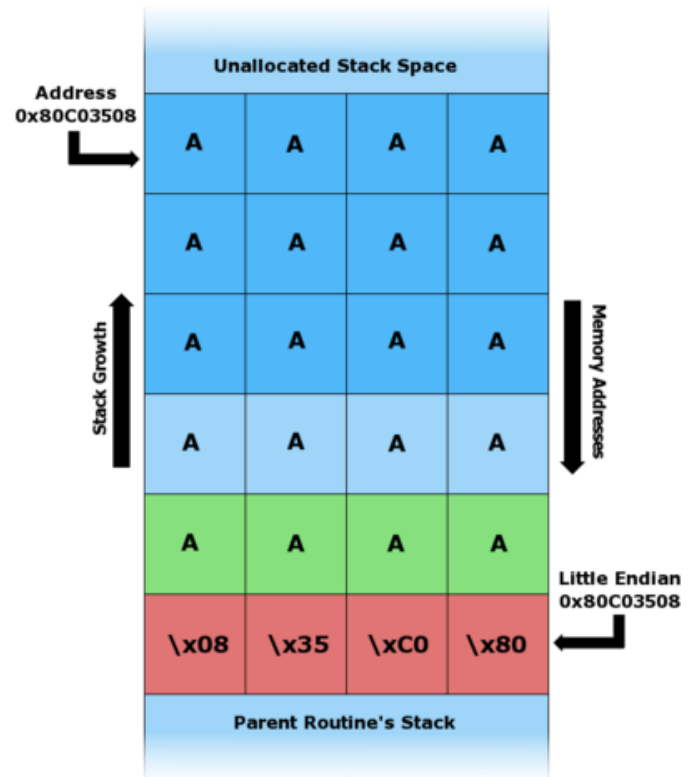


After passing  
"hello" on the  
command line

# Stack Buffer Overflow Example



Before



After passing "A A A A A A A A A A A A  
A A A A A A A \x08 \x35 \xC0 \x80" on the  
command line

# Stack Buffer Overflow Code Example with Arbitrary Code Execution

# Authentication

- We want users to access only accounts they're authorized to access.
- Controlling what a user can access on a system has two parts
  1. Authentication
  2. Authorization

# Authentication

- **Authentication** is verifying the identity of a party to a communication
- In some cases, such as an online bank, we want to authenticate both parties
- **Phishing** is the attempt to acquire sensitive information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity in an electronic communication.

# Authentication

- Usually takes one of three forms
  1. Something you have
    - House key, ATM card, RSA token
  2. Something you know
    - login or password
  3. Something you are
    - Fingerprint, voice print, retinal scan



# Two Factor Authorization

- Using two different methods of authentication at the same time is two-factor authorization
  - ATM card and PIN
  - Login plus RSA token



# Passwords

- Passwords are problematic because of social factors.
  - Worst is the default. Many times never changed.
  - weak password - easily guessed
  - strong password - combination of upper and lower case, symbols, numbers

# Passwords

- The easier a password is for the owner to remember generally means it will be easier for an attacker to guess.
- However, passwords which are difficult to remember may also reduce the security of a system because
  - users might need to write down the password
  - users will need frequent password resets
  - users are more likely to re-use the same password.
- Similarly, the more stringent requirements for password strength, e.g. "have a mix of uppercase and lowercase letters and digits" or "change it monthly", the greater the degree to which users will subvert the system.

# Dictionary Attack

- Passwords that are names or words can often be broken by guessing.
- Dictionary attack

# Worst Password Ideas

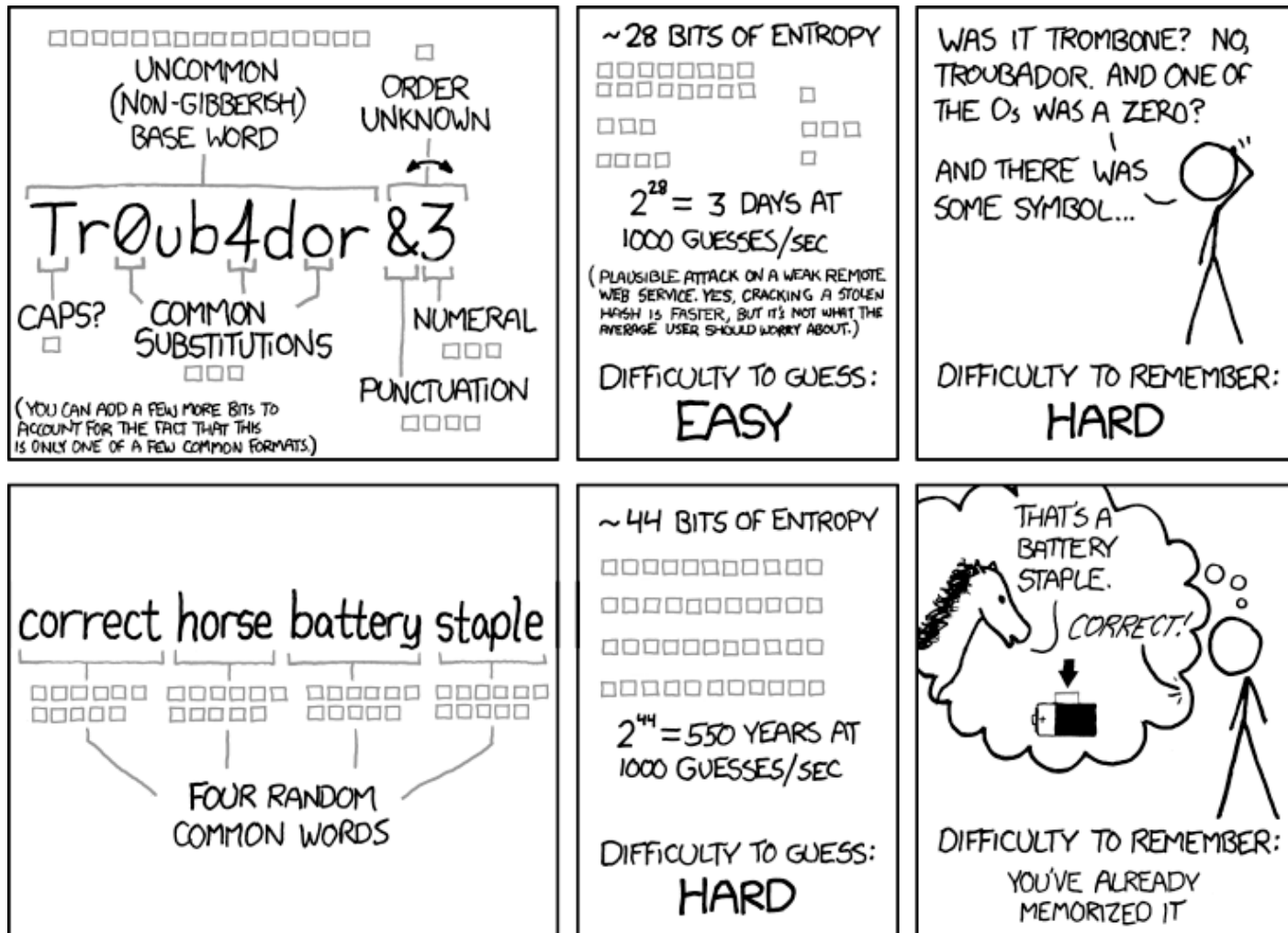
- Pet names
- A notable date, such as a wedding anniversary
- A family member's birthday
- Your child's name
- Another family member's name
- Your birthplace
- A favorite holiday
- Something related to your favorite sports team
- The name of a significant other
- The word "Password"

# Passwords

- In 2006, a survey of 34,000 MySpace passwords revealed that the most common were "password1", "abc123", "myspace1", and "password"

Password Authentication from a Human Factors Perspective: Results of a Survey among End-Users by  
Peter Hoonakker, Nis Bornoe and Pascale Carayon

# Passwords



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

# Authorization

- Once the OS knows who a user is, the next task is deciding what operations that user is allowed to do.
- Deciding what operation a user can perform is [authorization](#).
- In the context of an OS the “user” can be a process as well

# Access Control Matrix

- An abstract, formal security model of protection state in computer systems, that characterizes the rights of each subject with respect to every object in the system.
- An access matrix can be envisioned as a rectangular array of cells, with one row per subject and one column per object..



# Access Control Matrix

|        | gcc     | Mal's resumé  | Printer                            |
|--------|---------|---------------|------------------------------------|
| Mal    | Execute | Read<br>Write | Write                              |
| Jayne  | nil     | nil           | nil                                |
| Wash   | Execute | Read          | Write                              |
| Inara  | nil     | Read          | Write                              |
| Kaylee | Execute | Read          | Write<br>Stop Queue<br>Start Queue |

# Access Control List

- A list of permissions attached to an object
- More efficient than an ACM.
- List elements list only the users authorized to perform some action on the object.

| gcc    |         |
|--------|---------|
| Mal    | Execute |
| Wash   | Execute |
| Kaylee | Execute |

# Access Control List

- Access control lists still create more entries than we'd like.
- Instead use groups or roles.

# When do we check permissions?

- When a file is first accessed?
- Every operation?
- What if the user is in the middle of an operation when rights are revoked?

# Security and protection policies

- Any OS that is to be secure must have a set of policies that clearly spells out:
  - what users are allowed to do
  - what users are not allowed to do
  - what users are required to do
  - what the punishments are if the procedures are not followed

# Communication Security

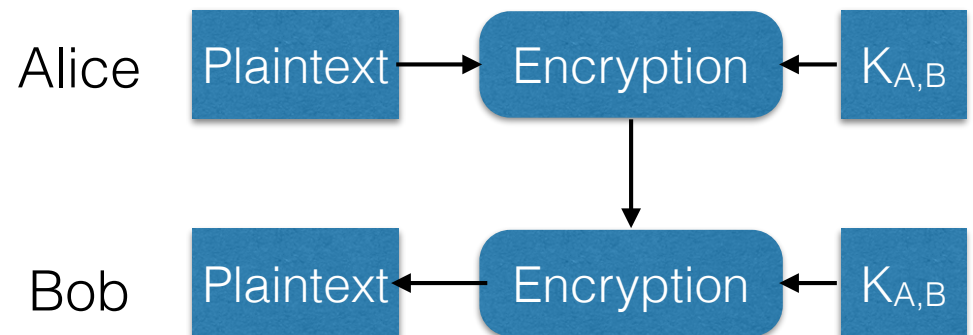
- Often a process running on one system will need to communicate with a process running on a different system.
- When we send information over the communication link three potential classes of problems can occur
  1. Intercepting
  2. Changing
  3. Inserting

# Encryption

- Encryption can eliminate or mitigate two out of the three problems.
- Encryption takes a message (plaintext) and uses a known algorithm to scramble the message.
- Relies on the fact that a captured message will be computationally infeasible to decrypt without knowing the key.
  - Theoretically can try every possible key value in a **brute force** attack.
- Unfortunately the definition of computationally infeasible changes.
  - What was computationally infeasible 5 years ago may be easy now.

# Symmetric Key Encryption

- In symmetric key encryption the same key is used to decrypt and encrypt the message.
- Algorithms that use this method are also known as shared key or secret key





# Symmetric Key Encryption

- Data Encryption Standard (DES) - Many years the standard but no longer considered secure.
  - 56 bit key
- In 2001 Advanced Encryption Standard (AES) was established.
  - 128, 192, or 256 bit key.
  - When AES was released DES could be broken in a few hours of brute force. \$10,000 of specialized hardware.
  - AES with a similar but faster machine would take 149 trillion years.

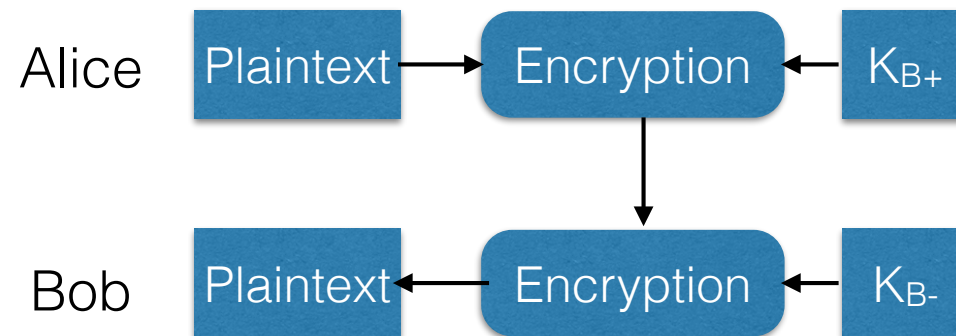
# Symmetric Key Encryption

- What happens if Bob and Alice don't know each other and are reluctant to share secret keys?
- Use a trusted third party (TTP) to generate a key and send it to both of them.

# Asymmetric Key Encryption

- Also known as public key encryption.
- Utilizes a pair of keys.
  - One key can be known to the world, the public key.
  - The other key, the private key, is kept secret.

# Asymmetric Key Encryption



# Asymmetric Key Encryption

- Standard for many years has been Rivest, Shamir, Adleman (RSA).
- Based on prime numbers
- Relies on the fact that there are efficient algorithms for determining if a number is prime, but no efficient algorithm for finding the prime factors of a number.

# Message digests

- Sometimes we don't necessarily want to hide the contents of a message we only want to make sure it didn't get changed.
- We can computer a message digest or hash.
  - Chop the message into short pieces and combine them in a one-way function.
  - The result is a message digest of fixed length, usually 128 bits

# Message digests

- Two algorithms are presently in use MD5 and secure hash standard (SHA).
- MD5 is commonly used to validate file downloads from the Internet.
- MD5 is breakable with only modest amounts of computing power. So only good for ensuring files were downloaded correctly.

# Message Signing

- By combining a message digest with public key encryption Alice can effectively sign a message electronically.
  - Alice takes message  $M$  and creates a digest.
  - She then encrypts the digest with her private key and sends the message and the encrypted digest to Bob.
  - Bob knows her public key so he can run the digest algorithm on the message and compare the result to the decrypted digest. If they are equal he knows Alice sent the message.



# Message Signing

- A special use of message signing is used to authenticate either clients or servers.
  - Process produces a certificate the verifies identity.
  - A special program is run that produces a preliminary certificate.
  - The preliminary certificate is sent to a certificate authority which encrypts the certificate with its private key.
  - Browsers can decrypt the certificate with the public key of the signing authority.

# Security Protocols

- What level in the network stack provides security?
  - Several
- In TCP/IP security is specified for the transport and network layer
- 802.11 there may be encryption at the data link layer.
- Transport layer security features are defined in the secure socket layer (SSL), also called transport layer security.
  - Also commonly used in application layer protocol HTTPS
- Network layer security also available with a protocol known as IPsec.