SLE 2018
Cyber Security

"Defense Against the Dark Arts"

LTC William Clay Moody, Ph.D.
madeye

Agenda

Morning Sessions

0900 - 0920 Introductions

0920 - 1020 Encoding and Encryption

1020 - 1030 Break

1030 - 1130 Web Security

1130 - 1300 Lunch

Afternoon Sessions

1300 - 1400 Reverse Engineering

1400 - 1410 Break

1410 - 1510 Binary Exploitation

1510 - 1530 Conclusion and Wrap Up

Introductions

Me:

```
4. bash
madeye at iMacEECS in ~
$ whois madeye
Name: LTC William Clay Moody

Position: Assistant Professor

Schools:
        BS,  Computer Engineering - Clemson Unversity
        MS,  Computer Networking  - North Carolina State University
        PhD, Computer Science     - Clemson University

Military:
        2LT - CPT: Signal Corps Officer
        CPT - MAJ: Telecommunication System Engineer
        MAJ - LTC: Cyber Warfare Officer

Teaching:
        Introduction to Programming
        Network Engineering and Management
        Operating Systems
        Defense Against the Dark Arts

Coaching:
        Cadet Competitve Cyber Team
        Cyber Defense Exercise
        NSA Cyber Exercise

Slogan:
        Constant Vigilance

madeye at iMacEECS in ~
$
```

You:

Full Name / Hacker Handle

Hometown / School

Hobbies / Person Interest

Prior Computer Experience

Fun Fact: High School "Capture the Flag" hacking competitions have taken off over the past few years!

Encoding and Encrypting

How computer store information and how we can protect our data

RIPPED FROM THE HEADLINES

WannaCry ransomware: what is it and how to protect yourself

The latest on the MS17-010 flaw and the WannaCry patch linked to the NHS cyber attack

By VICTORIA WOOLLASTON

Monday 22 May 2017

MAY 2017

Encoding: How computers store information

Binary: base-2 encoding

Series of 0s and 1s represent values

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

0 1 0 1 1 0 0 1

$2^6 + 2^4 + 2^3 + 2^0$

$64 + 16 + 8 + 1 = 89$

Fun Fact: a binary number is called a Bit... The USMA Cyber Team is called Bits For Everyone

Decimal: base-10 encoding

Series of digits from 0-9

10^3 10^2 10^1 10^0

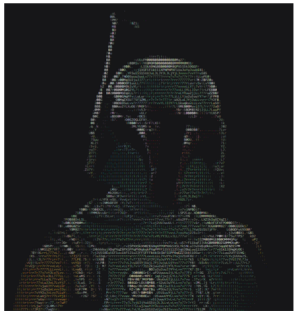
1 8 0 2

$1*10^3 + 8*10^2 + 2*10^0$

$1000 + 800 + 2 = 1802$

Fun Fact: United States Military Academy was founded in 1802

Advanced Encoding: Printable Characters



ASCII (American Standard Code for Information Interchange)

Pronounced like "Ask Key"

How to represent letters, numbers, and symbols digitally in a computer with a byte (8 bits)

Example:

'Y' is ASCII 89 / binary `0b01011001`
'@' is ASCII 64 / binary `0b01000000`
'1' is ASCII 49 / binary `0b00110001`
'c is ASCII 99 / binary `0b01100011`

Fun Fact: There are `10` types of people in the world: those that understand binary and those that do not.

Hexadecimal: base-16 encoding

Series of digits from 0-9, A-F

4 bits make one hex digit
ASCII Character represented with 2 hex digits

Example:

'Y' is hex `0x59`
'@' is hex `0x40`
'1' is hex `0x31`
'c' is hex `0x63`

Fun Fact: Hackers like to make phrases with hex code like `0xdeadbeef` and `0xc0deface`

Advanced Encoding: Data and Non-Printable

Base64: What it sounds like…

Series of symbols from A-Z, a-z, 0-9, /, +

256 possible bytes (2^8) but ASCII only goes to 127, many other possible bytes are not printable.

Take **3** bytes of **8** bits (24 total bits) and represent it as **4** symbols of **6** bits ($2^6 = 64$)

If the math does not work out (bytes are not multiple of 3) use the = as padding. Base64 message could have 1 or 2 equal signs at end

Fun Fact: Since 3 bytes are expanded to 4 bytes, Base64 encoding takes up more space that just the raw data

Example:

Binary stream: `11100011100000101000010110010110`

Byte Values: 227, 130, 133, 150

Hex Bytes: 0xE3, 0x82, 0x85, 0x96

Base64 groups: `111000 111000 001010 000101`

`100101 10**** ****** ******`

Base64 value: `44KFlg==`

Fun Fact: Seeing multiple equal signs at the end of message is a dead give away that it's Base64 encoded

Encryption: Keep our information secret

Rotational Encryption (ROT):

Letters of alphabet are replaced on a rotation.

Also called **Caesar Cipher**

Rotational amount can vary. Popular amounts are **3** and **13**.

To decrypt, rotate by 26 minus original rotation.

Example

Plain Text Message:

```
we attack at dawn
```

Rotation of 3:

```
zh dwwdfn dw gdzq
```

Rotation of 13:

```
jr nggnpx ng qnja
```

Fun Fact: The Orphan Annie Secret Society decoder pin in A Christmas Story used a simple caesar cipher to advertise ovaltine. Ralphie did not like that one bit.

Encryption: Protect your data

Substitution Cipher:

Like rotational, a substitution cipher replaces letters in a plain text message with other letters.

This time the mapping is not adjacent to each other.

Frequency analysis can help find mapping.

Like the cryptograms in the newspaper.

Example

Ciphertext:

```
whnq pehqm gjt pmamj lmgqp goh
```

Mapping Hint:

```
O = G
```

Plain Text:

```
four score and seven years ago
```

Fun Fact: The name "etaoin shrdlu" is composed of the top 12 most common letters in order. Characters and organizations with this name appears in many forms of media

Advanced Encryption:

Public Key Cryptography:

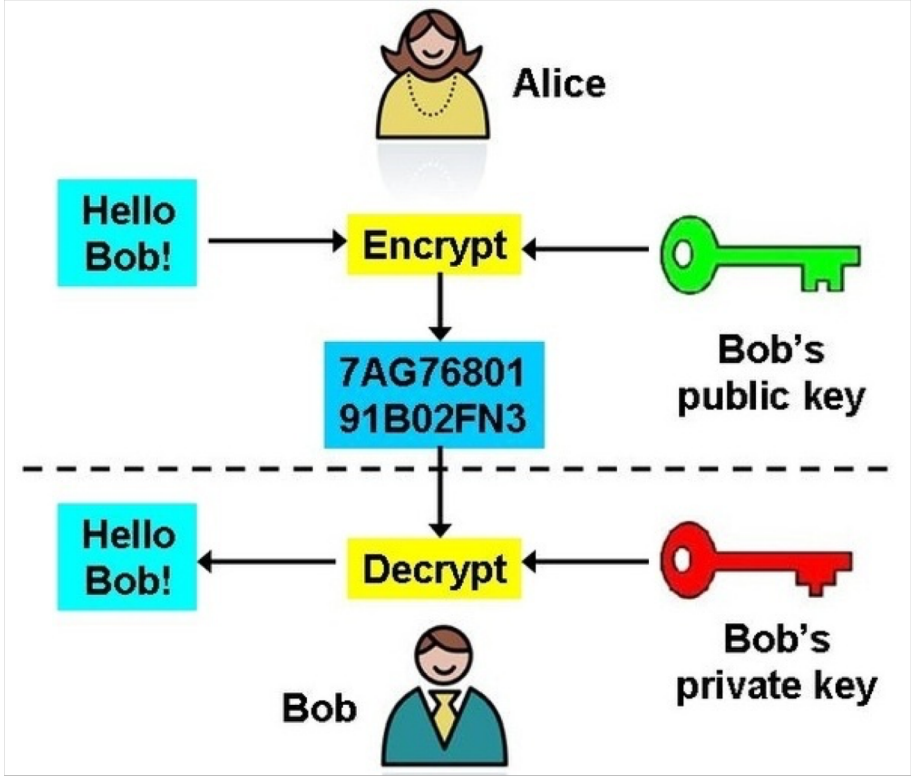
Two keys, one is **public** and the other is **private**.

User can **share their public key** with the world. But keeps **private key a secret**.

What is encrypted with the public key can only be decrypted with the private key

Even with the “locking” public key, you cannot “unlock” the message.

Advanced mathematical functions that cannot be reversed with specific information.



Advanced Encryption: RSA Algorithm

One of the first public key cryptosystems.

Takes advantage of factoring a number that is product of large prime numbers

Public Key is composed of two pieces of information: **N and e**

N is the product of two primes called **p and q**

e is the public exponent and is the inverse of the private exponent d.

p, q, and **d** are all kept secret

Modular Inverse:

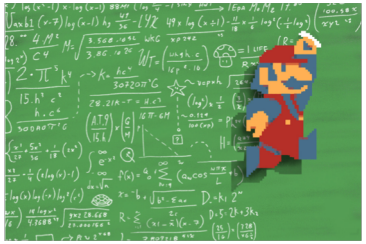
A special number **ϕ** is the product of **p-1** and **q-1**.

The remainder of e times d when divided by ϕ is always 1.

That is what it means to be an inverse.

Fun Fact: RSA is the initials of its three inventors: Ron Rivest, Adi Shamir, and Leonard Adleman

Advanced Encryption: RSA Algorithm



Example

Public key: N = 323; e = 11

Thus, **p** and **q** are 17 and 19

ϕ then would be 288

Taking every number between 1 and 288 and multiplying it by **e** then dividing it by **ϕ** will show us a remainder of 1 when d is 131

That is our **private key** d=131

Fun Fact: RSA Security company ran competitions for factoring numbers. The master challenge was a 2048 bit value. The prize money was $200,000.

How to use it

Encrypt: Take the ASCII value for each letter, raise it to the power of e then divide by N to get the remainder. The remainder is the cipher text for that letter

Decrypt: Take the ciphertext value and raise it to the power of d, then take remainder when dividing by N to get the plain text letter.

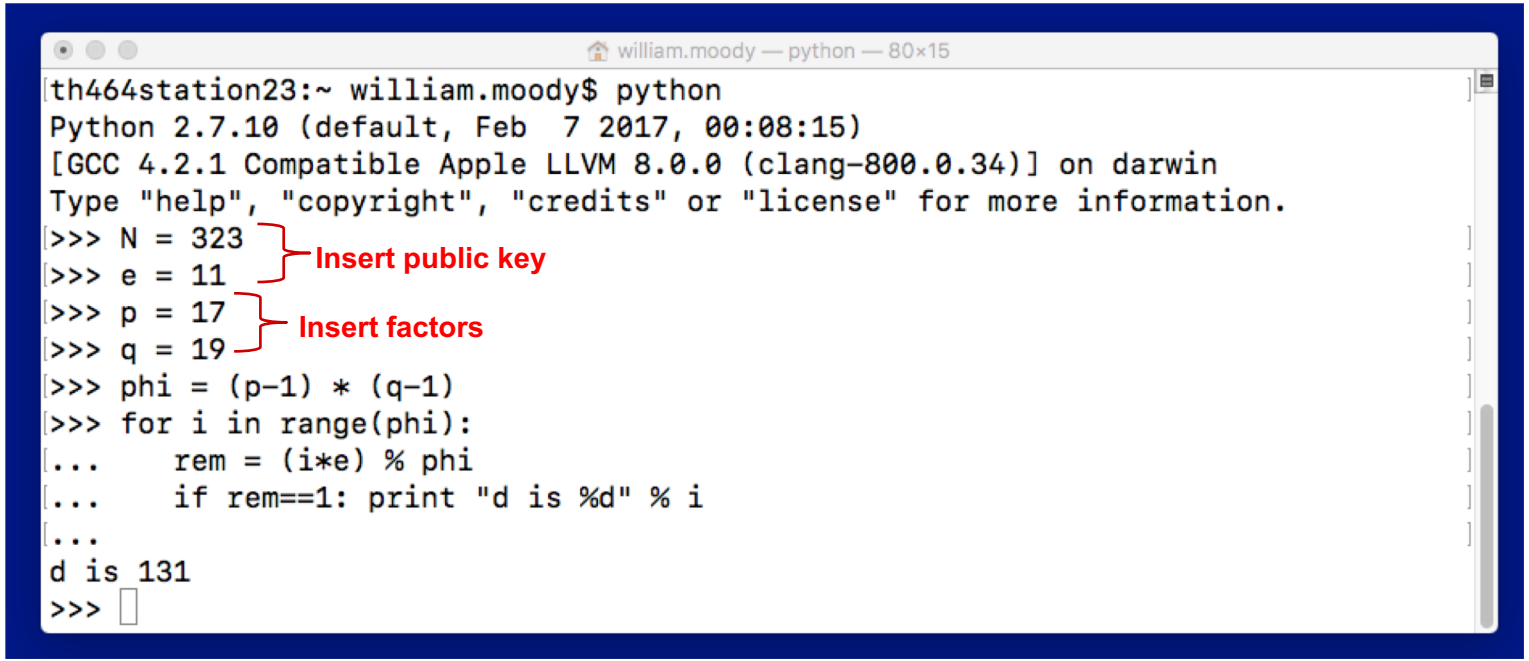
Plaintext message (m) = "**M**" [ASCII - 77]

$$77^{11} \bmod 323 =$$
$$5641543963891374499973 \bmod 323 = 134$$

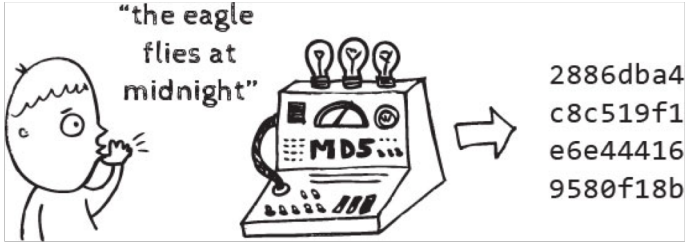
Ciphertext message (c) = 134

$$134^{131} \bmod 323 =$$
$$4.4474 * 10^{277} \bmod 323 = 77 \text{ (ASCII M)}$$

Python script for calculating inverse



Hashing: One way functions



A function that can be used to map data with an arbitrary size to data of fixed size.

This function cannot be reversed

Collisions are rare where two different inputs hash to the same output

Lookup tables exist where a hash input and output are stored.

Used to verify the integrity of the data since a single bit flip makes huge changes in the hash

Common hash algorithms include:

MD4

MD5

SHA1

SHA256

SHA512

Encoding and Encrypting Resources

ASCII Chart: https://www.asciitable.com/

Base64: https://www.base64encode.org/

ROT: https://tech.pookey.co.uk/non-wp/rot-decoder.php

Letter Frequency: http://letterfrequency.org/

Quip Quip: https://quipqiup.com/

RSA Site: https://goo.gl/xWEMFg

Cyber Chef: https://gchq.github.io/CyberChef/

Crack Hashes: https://crackstation.net/

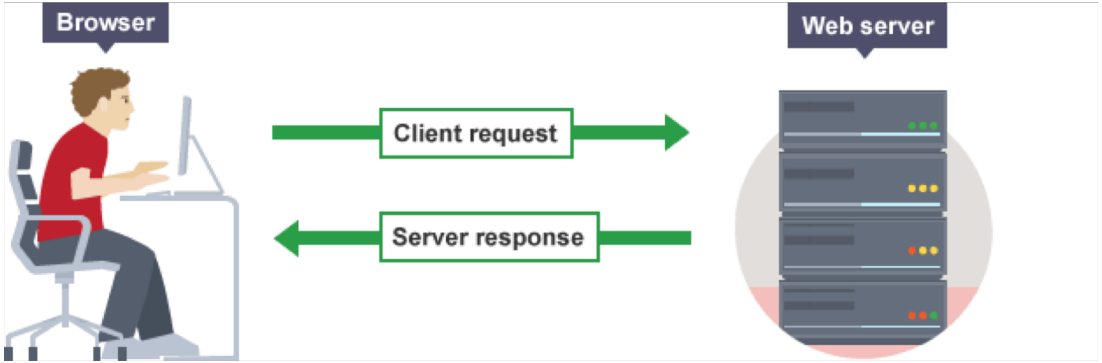
World Wide Web

Exploiting web servers and web clients

RIPPED FROM THE HEADLINES

SEPTEMBER 2017

Web: How Clients and Servers Talk

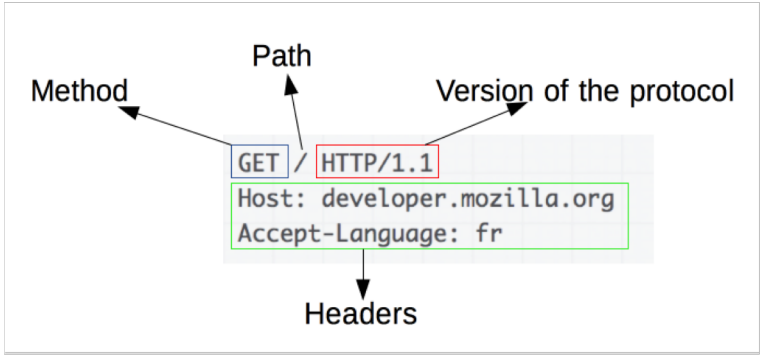


Hypertext Transport Protocol (HTTP): how browsers and clients talk

Hypertext Markup Language (HTML): how web pages are described

Fun Fact: Microsoft Internet Explorer is the world's number one browser for downloading other browsers

Web: Client Request Methods



HTTP verb used to talk to the webserver.

Most popular are GET and POST

GET: when visiting a website, your browser fetches the HTML code

POST: when logging in to a website or submitting data on a form

Other methods exist also:

- PUT
- HEAD
- DELETE
- TRACE
- OPTIONS
- CONNECT
- PATH

Fun Fact: As an April Fool's joke in 1998, someone defined the Hyper Text Coffee Pot Control Protocol, with method of BREW

Send a different HTTP Method

Mac OS X Terminal : 

Type: `curl -X <request_method> <url>`

```
th464station23:~ william.moody$ curl -X POST www.google.com
<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-wi
dth">
  <title>Error 411 (Length Required)!!1</title>
  <style>
```

Web: HTTP Status Codes

Code received from a web server by a client after making a request.

```
HTTP 1.1 GET http://www.google.co.uk

200 OK
Accept-Ranges: none
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Date: Tue, 24 Nov 2015 00:40:45 GMT
Expires: -1
```

Long list of codes, but the most interesting:

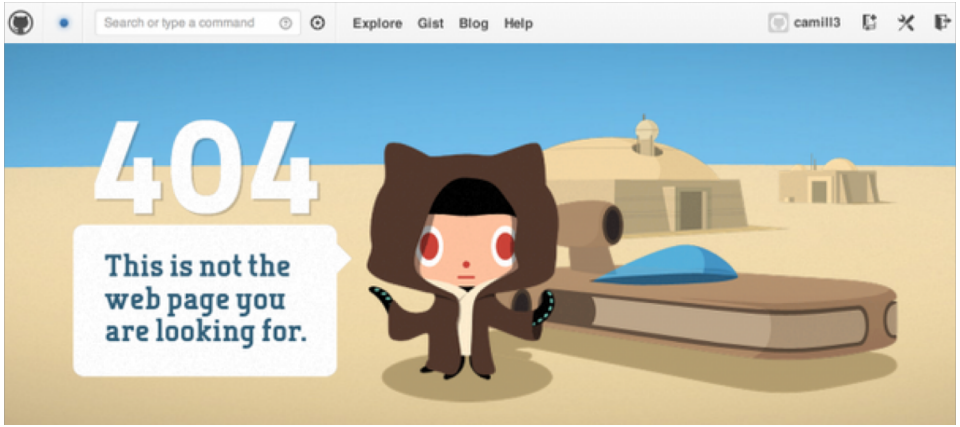
200: Success: OK

301: Redirection: Permanently Moved

302: Redirection: Found

404: Client Error: Not Found

500: Server Error: Internal Error



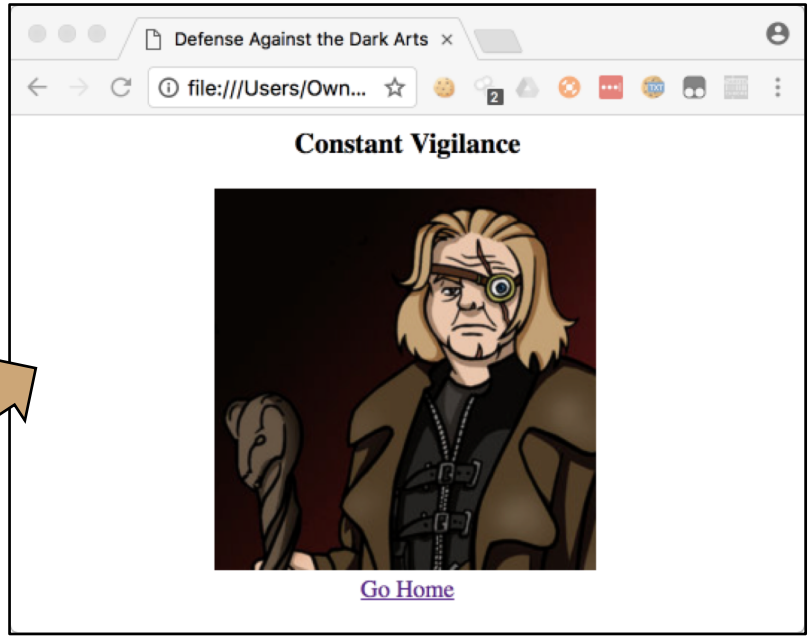
Fun Fact: Many sites have fun with their 404 pages. The current GitHub 404 is a Tatooine themed Star Wars joke.

Web: HTML - the language of the Web

Hypertext Markup Language is a well defined text description of a web page.

The browser knows how to translate the HTML into the website described.

```
<html>
    <head>
        <title>Defense Against the Dark Arts</title>
    </head>
    <body>
        <center>
        <h3>Constant Vigilance</h3>
        <img src='madeye.png'>
        <br />
        <a href="http://madeye.ninja">Go Home</a>
        </center>
    </body>
</html>
```



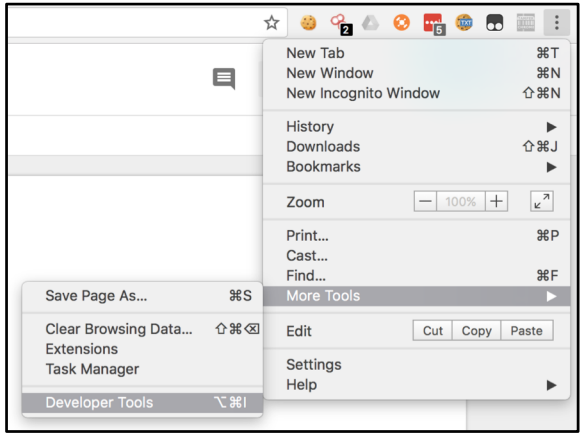
Fun Fact: Developers sometimes leave notes to themselves with comments like `<!-- this -->` These don't show on site.

Web: Javascript and CSS

Javascript:

Client-side code that runs in your browser. Builds dynamic sites and content.

Plaintext file is downloaded from server and rendered in your browser.



Cascading Style Sheets:

CSS files allow site wide formatting to be applied to HTML code. Separates data from view.

Many prebuilt packages available like Bootstrap

FUN FACT: YOU CAN INTERACT WITH IT IN THE CONSOLE UNDER THE DEVELOPER TOOLS

Web: Cookies, Cookies, Cookies

Cookies - how a website remembers who you are.

Used for legitimate reasons and not-so-legitimate reasons.

Good: log back into to a site you previously visited. Keep user custom settings. Shopping cart contents.

Bad: Targeted advertising based on search and browsing history. Track where you have been.

Fun Fact: Use the Developer Tools to see the COOKIES from your site (under Network and response)

Cookies can be edited with Developer tools or some plugins.

Web: Googlebot and other crawlers

Search engines and other automated systems crawl website to create maps of the world wide web.

A site can create a `robots.txt` file and host at the root of their web directory to dictate which parts of the site the bots can crawl.

Sites don't have to adhere to these guidances.

This is not a way to **protect** a site from visitors.

Formally called the *Robots exclusion standard*

https://www.starwars.com/rob ×

Roar May

Secure | https://www.starwars.com/robots.txt

```
# Robots.txt for www.starwars.com

User-Agent: *
Disallow: /7046/
Disallow: /products/
Disallow: /_xd/
Disallow: /news/page/
Disallow: /wp-content/plugins/
Disallow: /search
Disallow: /watch/hls/stream/
Disallow: /watch/hls/master/
Disallow: /watch/captions/

User-agent: dotbot
Disallow: /

User-agent: rogerbot
Disallow: /
```

Fun Fact: Google has protected their founders (Larry and Sergey) with a robots.txt file against various terminators

Web: Database Security

Accessing a website requiring a username and password typically involves a database.

Sending **POST** requests to a website with a login form normally run a **QUERY** on a database server.

The user provided input is used to build the QUERY asked of the database.

Trusting the user can be dangerous.

Example: a form provides the username (U) and password (P) variables.

The website puts the user provided strings into the following query:

```
SELECT * FROM users where username = 'U' and password = 'P'
```

A malicious user can enter a username that can is `' or 1=1 --` which will break the query. Since its always True, return all users.

Fun Fact: Geek humor talks about a boy named Little Bobby Tables from
https://xkcd.com/327/

Reverse Engineering

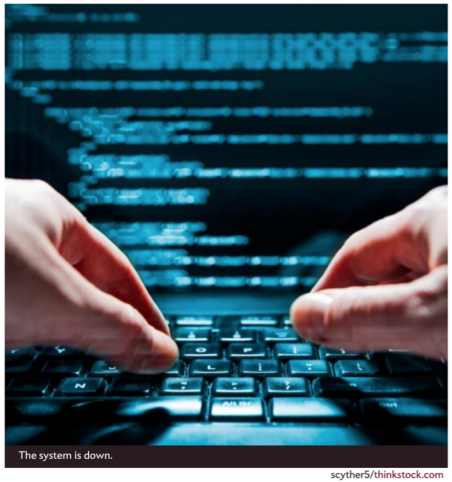
Discovering how software and hardware works so you can take advantage of it

RIPPED FROM THE HEADLINES

THE CITIZEN'S GUIDE TO THE FUTURE | OCT. 21 2016 3:09 PM

The East Coast Cyberattack: What We Know Now

By Jacob Brogan



The system is down.

scyther5/thinkstock.com

If you're located in the Eastern United States, odds are good that you've noticed that the internet is a little ragged today. On Friday morning, a **distributed denial of service attack** against the company Dyn brought down websites and apps across the internet, temporarily barring access to Twitter, Pinterest, WhatsApp, and more for millions of users. While Dyn was able to stabilize the situation within a few hours, a second DDoS attack began in the early afternoon, again disrupting services across the web.

OCTOBER 2016

Executable Files (aka Programs)

Information on a computer can be either:

Data

Instructions

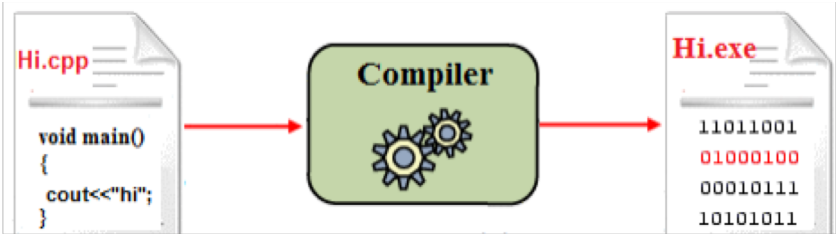
Examples:

Data - Movies, Images, Text Files, etc

Instructions - Programs, Operating System,

Files of instructions can be "executed"

How programs are built:



Data **Instructions**

FUN FACT: ON MAC, MAKE A FILE EXECUTABLE SO IT CAN BE "RUN" WITH THE FOLLOWING:

```
chmod +x filename
```

Python Programming Language

Created in 1989 by **Guido van Rossum**

Simple syntax, very powerful

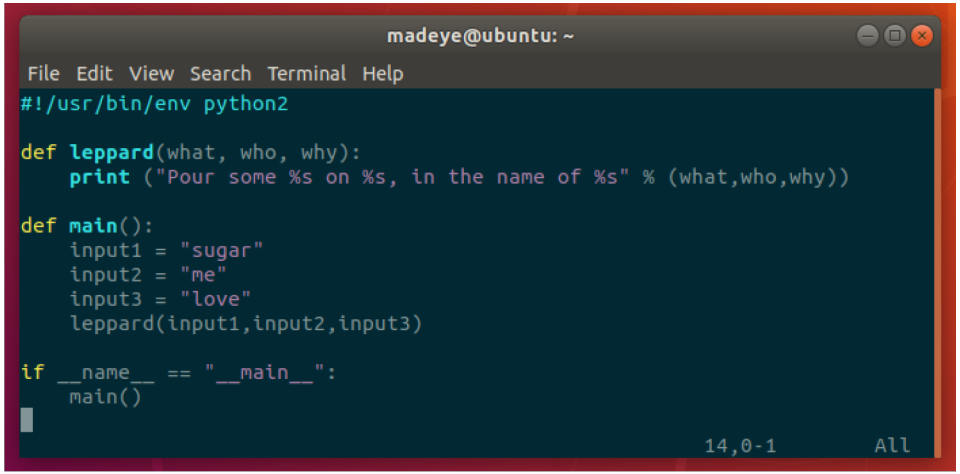
Tool of many a **hacker**

Taught to Plebes at USMA in IT105

White Space is very important!

Interpreted Language - meaning each command executed directly

Can be compiled in to `.pyc` files



Fun Fact: Python gets its name from Monty Python. Python Document is full of references to Holy Grail

Java Programming Language

Created by **James Gosling** in 1995 with Sun Microsystems (now Oracle)

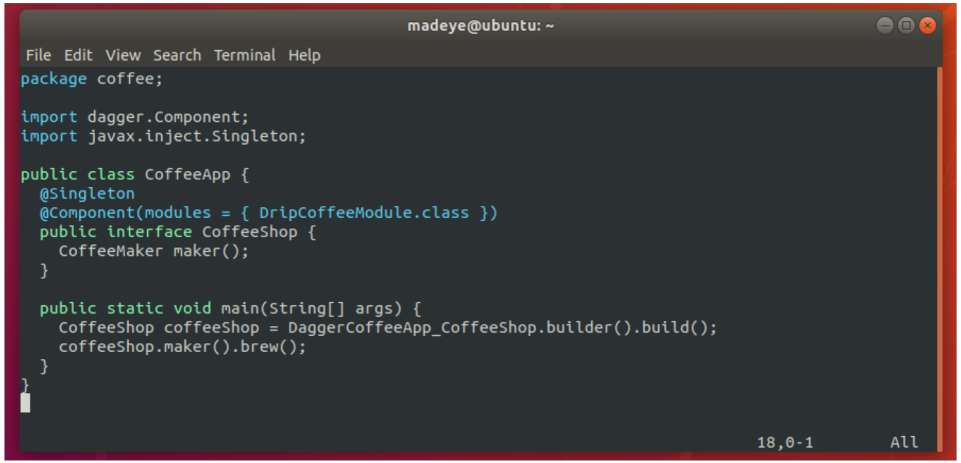
Write Once, Run Anywhere through use of a Virtual Machine

Android Operating system and most Android Apps are primary written in Java

Object Oriented Programming

Language of choice for AP Computer Science

Compiles to `.class` or `.jar` files



Fun Fact: Every Java Class file starts with these 4 bytes: `0xCAFEBABE` inspired after the coffee shop where Gosling regularly ate

C Programming Language

Designed **Dennis Ritchie** between 1969 and 1973 at Bell Labs

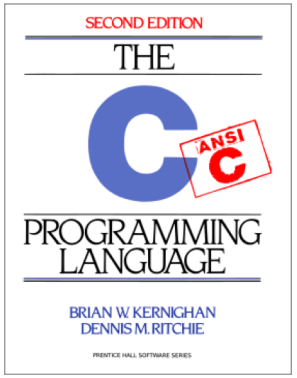
Unix and Linux operating system are written in C

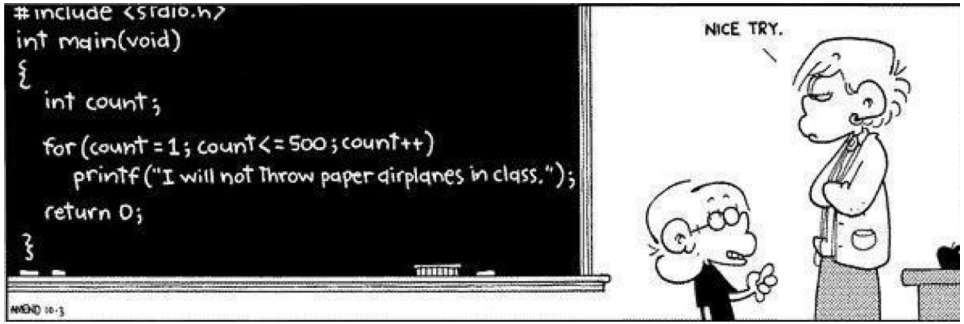
Much more low level that Python and Java

Better performance but very dangerous if not used properly

Recognized by its semicolons and curly braces

C-files compile to ELF files on Linux and Mach-O files on Mac





Fun Fact: The C Language does not do memory management well so you can crash a computer running C programs if not designed properly

What if you don’t have the source code?

If you have an executable file, how do you get the source code to know how it works?

Reverse Engineering is how you take a “binary executable” and discover how it works.

Many tools exist to help you “reverse” the compilation process.

Many people learn this skill to use unauthorized software. “License key cracks”

*This is **not** inline with the values of the Army or West Point.*

Tools of the trade

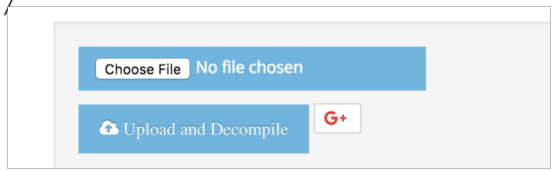
Command Line Tools:

`strings <file_name>`

View printable characters in a file (all types)

Online tools:

`http://www.javadecompilers.com/`



Binary Ninja:

`Installed on the Macs`

Uncompyle6:

Use with the following: `uncompyle6 filename`

(Installed on the shell server, ssh from mac)

```
cadet@th464station9:~$ ssh sle01@sle.c3t.eecs.net
Enter your password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Sat Jun  2 12:39:51 2018 from 10.64.0.101
sle01@sle:~$ uncompyle6 rev_me.pyc
```

Arrays of Characters

In programming, sets of identical data types are called **Arrays**

When you have an array of characters that is called a **String**

When you want to access a specific character in a string, you can do that with an array **index**. This is the position in the array (starts at 0)

When you want a subset of a string, that is called a **substring** or a **splice**

Python

```
madeye@ubuntu:~/samples$ cat pystrings.py
#!/usr/bin/env python

message = 'Summer Leadership Training'
print message
print message[5]
print message[3:12]
madeye@ubuntu:~/samples$ python pystrings.py
Summer Leadership Training
r
mer Leade
```

Java

```
madeye@ubuntu:~/samples$ cat jstrings.java
public class jstrings {

    public static void main(String[] args) {
        String message = "United States Army";
        System.out.println(message);
        System.out.println(message.substring(5,6));
        System.out.println(message.substring(8,15));
    }
}
madeye@ubuntu:~/samples$ java jstrings
United States Army
d
tates A
madeye@ubuntu:~/samples$
```

C

```
madeye@ubuntu:~/samples$ cat cstrings.c
#include <stdio.h>
#include <string.h>

int main (int argc, char *argv[]) {
        char *msg = "Hack the Planet";
        char substring[5] = { 0 };
        printf("%s\n", msg);
        printf("%c\n", msg[6]);
        memcpy(substring,msg+9,4);
        printf("%s\n",substring);
}
madeye@ubuntu:~/samples$ ./cstrings
Hack the Planet
h
Plan
```

Hacker Tips:

In the Mac Terminal:

Change Directories: `cd <dir_name>`

List Contents: `ls`

Identify File: `file <file_name>`

Make Executable: `chmod +x <file_name>`

Run a program: `./file_name`

Binary Exploitation

My other computer is your computer

RIPPED FROM THE HEADLINES

The Washington Post (WP Company

NSA officials worried about the day its potent hacking tool would get loose. Then it did.

The National Security Agency campus in Fort Meade, Md. (2013 photo by Patrick Semansky/AP)

By Ellen Nakashima and Craig Timberg May 16, 2017 Email the author

When the National Security Agency began using a new hacking tool called EternalBlue, those entrusted with deploying it marveled at both its uncommon power and the widespread havoc it could wreak if it ever got loose.

MAY 2017

Binary Exploitation

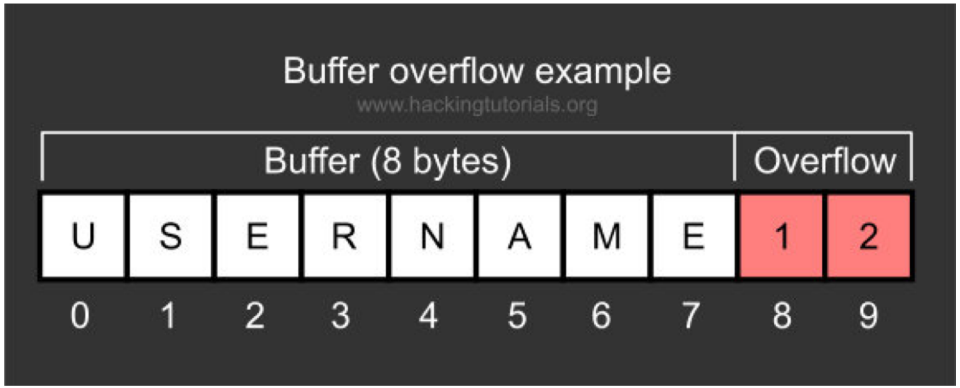
Many ways to break a binary!

We will mainly talk about one of the oldest forms of exploitation.

Pretty technical, hopefully you will see the impact.

We will talk about:

Buffer Overflow



Fun Fact: Buffer Overflows were introduced in a paper by Aleph One in Phrack Magazine called "Smashing the Stack for Fun and Profit"

The Stack

The modern computer architecture uses a **stack** to manage memory.

A stack is a “**Last In, First Out**” data structure. Like a stack of dishes or trays.

Each function has its on “stack” of memory. A function that called from another function, puts its stack on top of the caller.

Data and Instructions are both on the stack

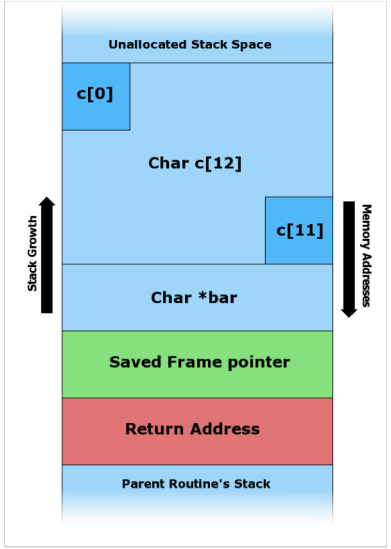
Stacks are limited in size and a buffer overflow that smashes the stack can result in bad things!

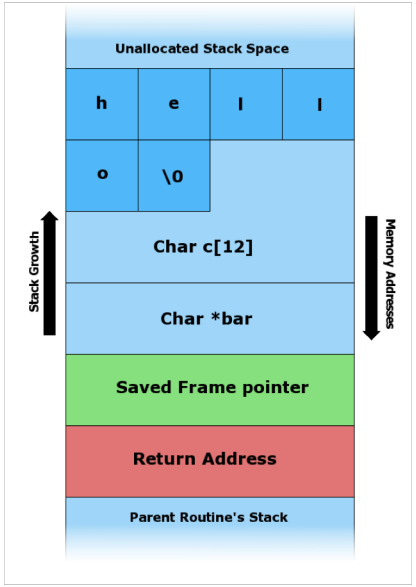
Example:

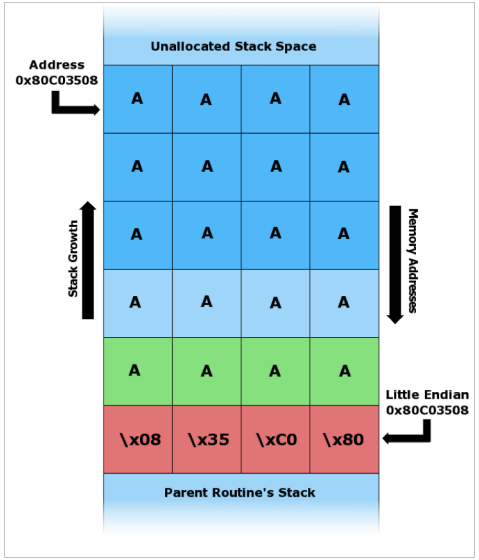
```
madeye@ubuntu: ~/samples
File  Edit  View  Search  Terminal  Help
#include <string.h>

void foo (char *bar)
{
   char  c[12];
   strcpy(c, bar);  // no bounds checking
}

int main (int argc, char **argv)
{
   foo(argv[1]);
   return 0;
}
                                                14,0-1        All
```







What's the worst that can happen?

Crash the program. What if it's a critical process?

Fill the stack with executable code, make the function return call execute code on the stack

Overwrite important data. Change your bank account?

Overwrite the return address and run other code

Hacker Tips:

Interacting with services:

In the Mac Terminal: 

Type:

```
nc <provided_host> <provided port>
```

When landing a shell:

List the directory:

Type:

```
ls
```

To see contents of file:

```
cat flag.txt
```