Secure Programming

Mehmet Tahir SANDIKKAYA

Spring 2022

Istanbul Technical University
Computer Engineering Department

Syllabus

Week	Date	Rct	Covers	Subject	Announcement	Submit
1	22nd Feb		7	Fundamental concepts of security	TP-A	
2	01st Mar		1, 7	Compilation and Execution		
3	08th Mar		1, 7	Stack overflow and its mitigation	Asg1-A	
4	15th Mar		1, 7	Dynamic memory management		
5	22nd Mar	R	3, 7	Canonicalization attacks and mitigation	Asg2-A	Asg1-S
6	29th Mar		2, 7	Injection attacks	Asg1-G	
7	05th Apr	R	2, 7	Injection mitigation	Asg3-A	Asg2-S
8	12th Apr		4	Reverse engineering and obfuscation	Asg2-G	
9	19th Apr		5	Fundamental cryptography		Asg3-S
A	26th Apr	R	5	Principles of computer communication	Asg3-G	MT
-	03rd May			Spring break		
B	10th May		5	XSS & CSRF attacks and mitigation	Asg4-A	
C	17th May	R	5	Race conditions	MT-G	
D	24th May		6	Permission and authorization mechanisms in contemporary languages		Asg4-S
E	31st May		7	Test and static analysis tools	Asg4-G	TP-S

Recitation

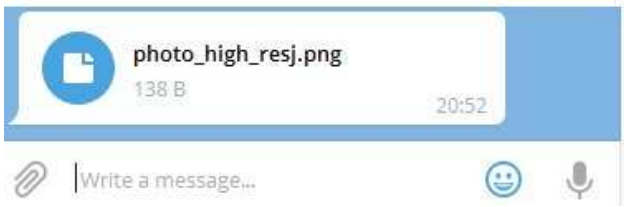
Stack Overflow

Recitation by Ayşe SAYIN covering stack overflows.

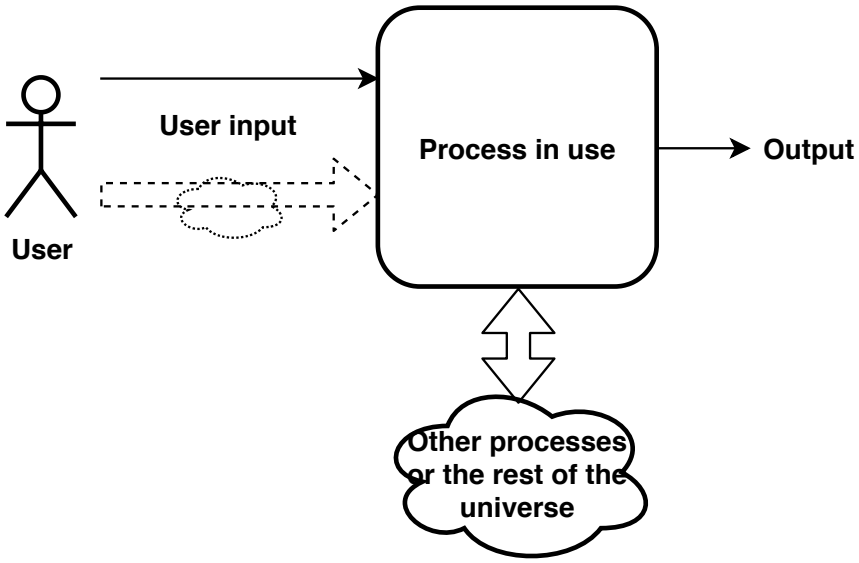
Canonicalization

What's your name?

- ✔ Normal: cannot be rewritten any further Mehmet Yılmaz
- ✔ Canonical: in its simplest or standard form 1312 1312 666
- ✔ The standard, most direct, and least ambiguous way of representation
- ✔ Canonicalization is the process by which various equivalent forms of a name are resolved to a single, standard name, which is called the *canonical* name
- ✔ Will you check the following file?



How could we communicate?



What should be the steps for trustworthy communication?

1. Refer to a resource
2. Retrieve correctly
3. Check validity
4. Interpret semantics

How to canonicalize?

Rewrite the initial resource specifier (if required, several times) till you have the unique specifier value and not an expression

Which type of language is this?

Context-free, therefore it could be defined as a series of rewrites.

Examples

- ✔ Java's `getCanonicalPath()` or POSIX `realpath()` function
- ✔ URI vs. URL
- ✔ Eager or lazy initialization of variables
- ✔ LaTeX's `\relax` or `\expandafter` (See [tex.stackexchange, 2012], [Wikibooks, 2011])

```
\def\a[#1]{A's argument is `#1'}
\def\args{[FOO]}
\a\args
\expandafter\a\args
```

```
\def\mysize{144pt\relax}
lorem ipsum...
```


Resource naming

- ✔ Cases
- ✔ Windows naming
- ✔ Symbolic links with rights
- ✔ NTFS data streams
- ✔ Systems, unfortunately, tend to fix your errors
- ✔ Unicode filename extension \\?\
- ✔ Directory traversal
- ✔ UNC shares
- ✔ Is your file really a file?

A side note on UTF-8

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 3
1	7	U+0000	U+007F	0xxxxxxx	-	-	-
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx	-	-
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	-
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

A side note on UTF-8

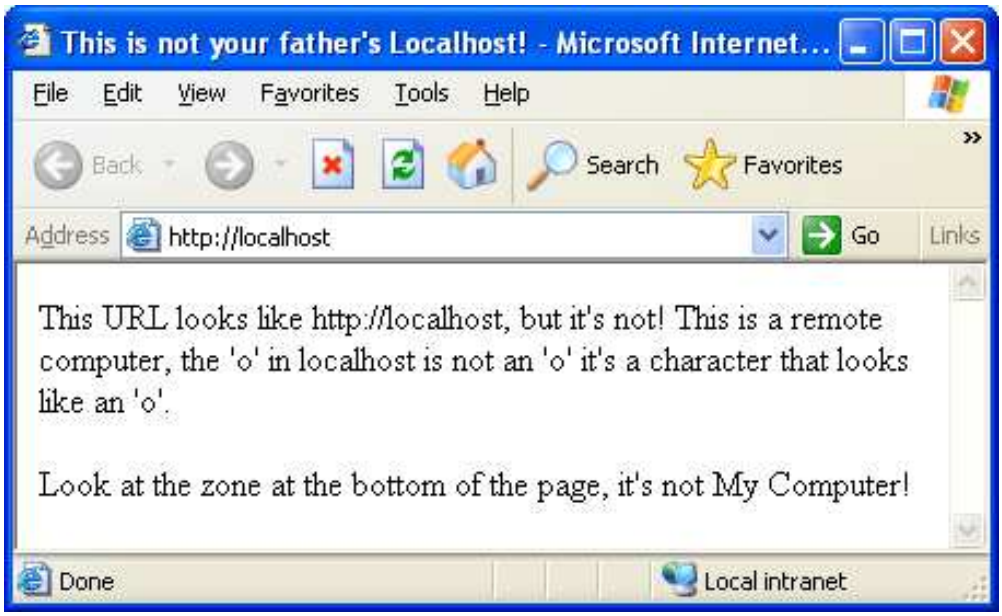
?

- ✔ 0x3F 0011 1111
- ✔ 0xC0 0xBF 1100 0000 1011 1111
- ✔ 0xE0 0x80 0xBF 1110 0000 1000 0000 1011 1111
- ✔ 0xF0 0x80 0x80 0xBF
- ✔ 0xF8 0x80 0x80 0x80 0xBF
- ✔ 0xFC 0x80 0x80 0x80 0x80 0xBF

Resources on the web

- ✔ When a line is not a line?
- ✔ `dummyFile.txt\r\nVicky\t13:03:00\tevilPlans.txt`
- ✔ `http://www.site.com/secretFile%2Etxt`
- ✔ `http://www.site.com/process.asp?file=../../winnt/repair/sam`
- ✔ `http://2689309198`
- ✔ `http://www.site.com/scripts/..%c0%af../winnt/system32/`
- ✔ Escaping is not your friend
  - ✘ %5c
  - ✘ %255c
  - ✘ %25%35%63

Visual representations



[Howard and LeBlanc, 2003]

How to avoid?

- ✔ Always limit inputs
- ✔ Do not make decisions based on resource names
- ✔ Use fully-qualified resource names
- ✔ Be careful when dealing with UTF-8
- ✔ Be careful of the time of referring a resource and using a resource
- ✔ Regular expressions may help a lot but only in regular languages (E.g. not HTML [Stackoverflow, 2014])

Do these till next week...

- ✔ Check examples on [Howard and LeBlanc, 2003]
- ✔ Please, get used to LaTeX (simply I cannot stand people who cannot typeset)

Bibliography

References

[Howard and LeBlanc, 2003] Howard, M. and LeBlanc, D. (2003). *Writing Secure Code*. Microsoft Press, 2nd edition.

[Stackoverflow, 2014] Stackoverflow (2014). RegEx match open tags except XHTML self-contained tags. `https://stackoverflow.com/questions/1732348/regex-match-open-tags-except-xhtml-self-contained-tags/`.

[tex.stackexchange, 2012] tex.stackexchange (2012). What is the difference between \relax and {}? `https://tex.stackexchange.com/questions/86385/what-is-the-difference-between-relax-and%7B/%7D%7D`.

[Wikibooks, 2011] Wikibooks (2011). TeX/expandafter. `https://en.wikibooks.org/wiki/TeX/expandafter`.