Messing with unicode

Julin Shaji

February 2022

Python devroom, FOSDEM 2022

Intro

- *Trojan Source: Invisible Vulnerabilities* - N Boucher, R Anderson
- PEP 672: *Unicode-related Security Considerations for Python*

Outline

Glyphs

Bidi

Mitigation

Glyphs

Glyphs

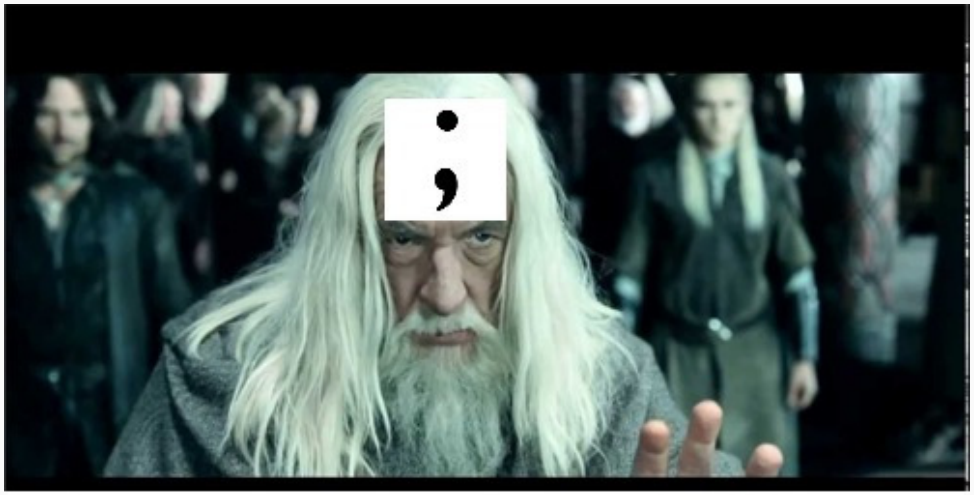
Bidi

Mitigation

Homoglyphs


REPLACE SEMICOLONS (;)
WITH GREEK QUESTION MARKS (;)
AND WATCH THE PROGRAMMER GO CRAZY
That's the evilest thing I can imagine.
imgflip.com
GIFSec.com

Homoglyphs



Homoglyphs

An example

```
>>> 'H' = 3
```

An example

```
>>> 'H' = 3
```

U+1FBF GREEK PSILI

An example

```
>>> ’H’ = 3
>>> ’H’ = 3
```

U+1FBF GREEK PSILI

U+0027 APOSTROPHE

An example

```
>>> ’H’ = 3
>>> ’H’ = 3
>>> ‘H‘ = 3
```

U+1FBF GREEK PSILI

U+0027 APOSTROPHE

U+02BB MODIFIER LETTER TURNED COMMA

Homoglyphs

;	(Greek ?)	;	(semicolon)
с	(Cyrillic)	c	(Latin)
∀	(forall)	∀	(inverted A)
丌	(CJK radical)	丌	(CJK unified)
ഩ	(Tamil e)	ഩ	(Tamil 7)
ﻁ	(Arabic TAH)	ﻁ	(Arabic TAH isolated)
и	(Deseret)	и	(Cyrillic)

Homoglyphs: Examples

```
def toss(face : str) -> str:
    if face == ”head”:
        return ”Alice”
    return ”Bob”
```

Homoglyphs: Examples

```
def toss(face : str) -> str:
    if face == "h е ad":     # Cyrillic e!
        return "Alice"
    return "Bob"
```

Homoglyphs: Examples

```
def foo():
    print(”Hi!”)

foo()
```

Homoglyphs: Examples

```
def f o o():      # Cyrillic o!
    print(”Hi!”)

foo()    # foo undefined!
```

Invisible characters

```
>>> "hello" == "hello"
```

Invisible characters

```
>>> "hello" == "hello"
```

False

Invisible characters

```
>>> "hello" == "h<ZWS>ello"
```

False

Invisible characters: Example

```
def toss(face : str) -> str:
    if face == ”head”:
        return ”Alice”
    return ”Bob”
```

Invisible characters: Example

```
def toss(face : str) -> str:
    if face == "h <ZWS> ead":     # Zero-width-space!
        return "Alice"
    return "Bob"
```

Invisible characters

U+200B	Zero Width Space
U+200C	Zero Width Joiner
U+200D	Zero Width Non-joiner
U+2060	Word joiner
U+FEFF	Zero Width No-break Space
U+2063	Invisible separator

Unicode normalization

```
>>> xⁿ = 5
>>> print(xn)
```

Unicode normalization

```
>>> xⁿ = 5
>>> print(xn)
Error??
```

Unicode normalization

```
>>> xⁿ = 5
>>> print(xn)
5
```

Unicode normalization

```
>>> xⁿ = 5
>>> print(xn)
5
>>> print(xₙ)
```

Unicode normalization

```
>>> xⁿ = 5
>>> print(xn)
5
>>> print(xₙ)
5
```

Unicode normal forms

Ç

Ç

Unicode normal forms

ç
U+00e7

ç
U+0063 + U+0327

U+00e7	LATIN SMALL LETTER C WITH CEDILLA

U+0063 U+0327	LATIN SMALL LETTER C COMBINING CEDILLA

Bidi

Glyphs

Bidi

Mitigation

Bracket direction

a < b < c

ا < ب < ت

א < ב < ג

Bracket direction

a < b < c

ا < ب < ت

א < ב < ג

a < b < c

ا > ب > ت

א > ב > ג

Writing direction

```
ب = 3
א = ب
ب - (א ** 2)
```

Writing direction

```
ב = 3
```

```
3 = ב
ב = א
(2 ** א) - ב
```

Writing direction

```
ב = 3
א = ב
```

```
3 = ב
ב = א
(א ** 2) - ב
```

Writing direction

```
3 = ב
ב = א
(2 ** א) - ב
```

```
ב = 3
א = ב
ב - (א ** 2)
```

Writing direction

```
ב = 3
א = ב
ב - (א ** 2)
```

```
3 = ב
ב = א
(א ** 2) - ב
```

Writing direction

```
ב = 3
א = ב
ב - (א ** 2)
```

```
3 = ב
ב = א
(א ** 2) - ב
```

Writing direction

```
ב = 3
א = ב
ב - ( א ** 2)
```

```
3 = ב
ב = א
(2 ** א ) - ב
```

Writing direction

```
3 = ב
ב = א
(2 ** א) - ב
```

```
ב = 3
א = ב
ב - (א ** 2)
```

Writing direction

```
ב = 3
א = ב
ב - (א ** 2)
```

```
3 = ב
ב = א
(2 ** א) - ב
```

Writing direction

```
3 = ב
ב = א
(2 * * א) - ב
```

```
ב = 3
א = ב
ב - (א * * 2)
```

Writing direction

```
3 = ב
ב = א
(2 ** א) - ב
```

```
ב = 3
א = ב
ב - (א ** 2)
```

Writing direction

```
3 = ב
ב = א
( 2 ** א) - ב
```

```
ב = 3
א = ב
ב - (א ** 2 )
```

Writing direction

```
a = "ا" # 5 * "ب"
```

Writing direction

```
a = "ا" # 5 * "ب"
'ببببب'
```

Writing direction

```
a = "|" # 5 * "ب"
'ببببب'
```

```
a = "ب" * 5 # "|"
```

Writing direction

```
a = "ا" # 5 * "ب"
'ببببب'
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ب" * 5 # "ا"
→-→ ←-←-←-←-←-←-←-
```

Writing direction

a

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a =
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا"
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا" #
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا" # 5
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا" # 5 *
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "|" # 5 * "
```

```
a = "ب" * 5 # "|"
```

Writing direction

```
a = "ا" # 5 * "ب
```

```
a = "ب" * 5 # "ا"
```

Writing direction

```
a = "ا" # 5 * "ب"
```

```
a = "ب" * 5 # "ا"
```

Bidi control chars

LRE	U+202A	Left-to-right embedding
RLE	U+202B	Right-to-left embedding
LRO	U+202D	Left-to-right override
RLO	U+202E	Right-to-left override
LRI	U+2066	Left-to-right isolate
RLI	U+2067	Right-to-left isolate
PDI	U+2069	Pop directional isolate
PDF	U+202C	Pop directional formatting

File names

```
annexe.txt

annexe.doc
```

File names

annexe.txt	ann<RLO>txt.exe
annexe.doc	ann<RLO>cod.exe

annexe.txt

a nn<RLO>txt.exe a

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex
ann<RLO>txt. e xe	annexe

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex
ann<RLO>txt. e xe	annexe
ann<RLO>txt . exe	annexe.

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex
ann<RLO>txt. e xe	annexe
ann<RLO>txt . exe	annexe.
ann<RLO>tx t .exe	annexe.t

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex
ann<RLO>txt. e xe	annexe
ann<RLO>txt . exe	annexe.
ann<RLO>tx t .exe	annexe.t
ann<RLO>t x t.exe	annexe.tx

annexe.txt

a nn<RLO>txt.exe	a
a n n<RLO>txt.exe	an
an n <RLO>txt.exe	ann
ann <RLO> txt.exe	ann<RLO>
ann<RLO>txt.ex e	anne
ann<RLO>txt.e x e	annex
ann<RLO>txt. e xe	annexe
ann<RLO>txt . exe	annexe.
ann<RLO>tx t .exe	annexe.t
ann<RLO>t x t.exe	annexe.tx
ann<RLO> t xt.exe	annexe.txt

Mitigation

Glyphs

Bidi

Mitigation

Code review tools

`f1.txt`:

```
{
```

`f2.txt`:

```
<RLO>}
```

Code review tools

`f1.txt:`

```
{
```

`f2.txt:`

```
<RLO>}
```

`diff f1.txt f2.txt`

```
1c1
< {
---
> }
```

Code review tools

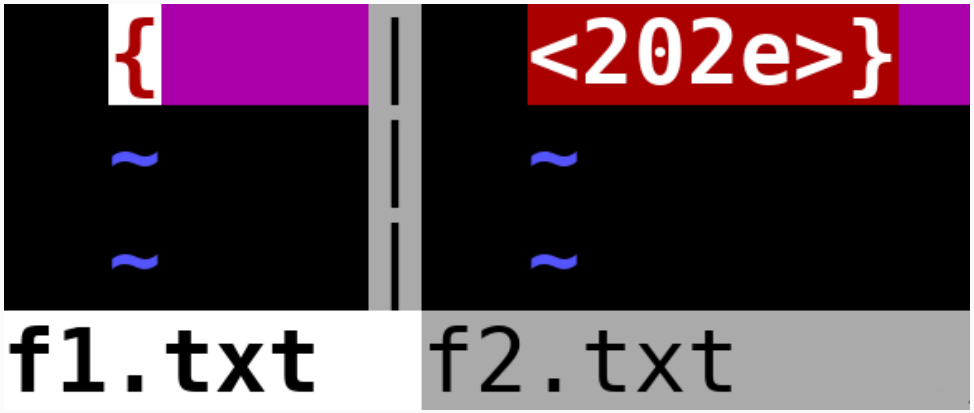
`f1.txt`:

```
{
```

`f2.txt`:

```
<RLO>}
```

Vim 8.0: `vimdiff f1.txt f2.txt`



Code review tools

`f1.txt`:

```
{
```

`f2.txt`:

```
<RLO>}
```

emacs 27.2: `M-x diff-buffers`

```
--- #<buffer f1.txt>
+++ #<buffer f2.txt>
@@ -1 +1 @@
-{
+{
```

Code review tools

`f1.txt`:

```
{
```

`f2.txt`:

```
<RLO>}
```

emacs -nw: `M-x diff-buffers`

```
--- #<buffer f1.txt>
+++ #<buffer f2.txt>
@@ -1 +1 @@
-{
+{
```

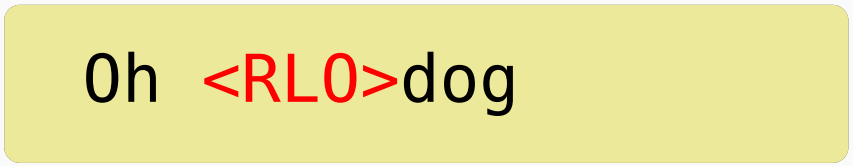
Cursor movement

```
Oh god
```

Cursor movement

```
Oh <RLO>dog
```

Cursor movement



Cursor movement

```
0h <RLO>dog
```

```
0h god
```

Cursor movement

```
Oh <RLO>dog
```

```
Oh god
```

Cursor movement

```
Oh <RLO>dog
```

```
Oh god
```

Cursor movement

```
Oh <RLO>dog
```

```
Oh god
```

Cursor movement

```
Oh <RLO>dog
```

```
Oh god
```

Cursor movement

```
Oh <RLO>dog
```

```
Oh god
```

Cursor movement

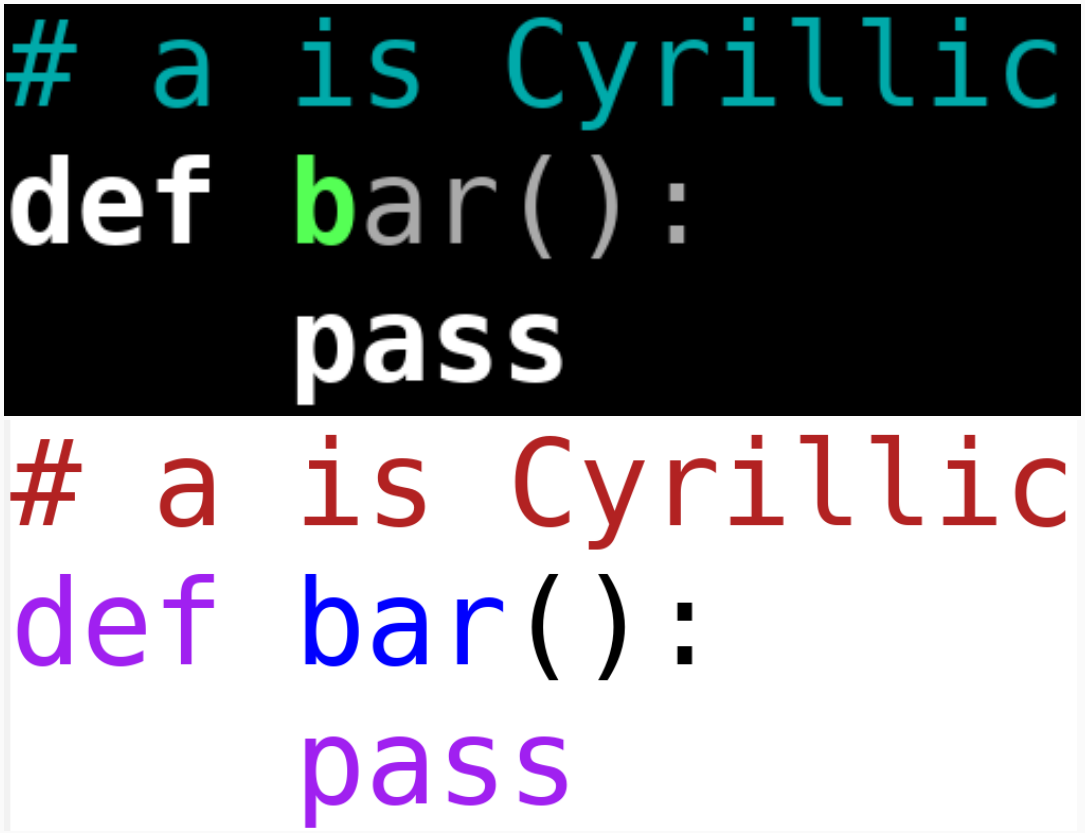
```
Oh <RLO>dog
```

```
Oh god
```

Syntax highlighting

```
# а is Cyrillic
def bar():
    pass
```

Syntax highlighting



Source forges

- Gitea
- Gitlab
- Github
- Bitbucket

References

- About Python identifiers
- A list of homoglyphs

Thanks!

Thanks!

Questions?