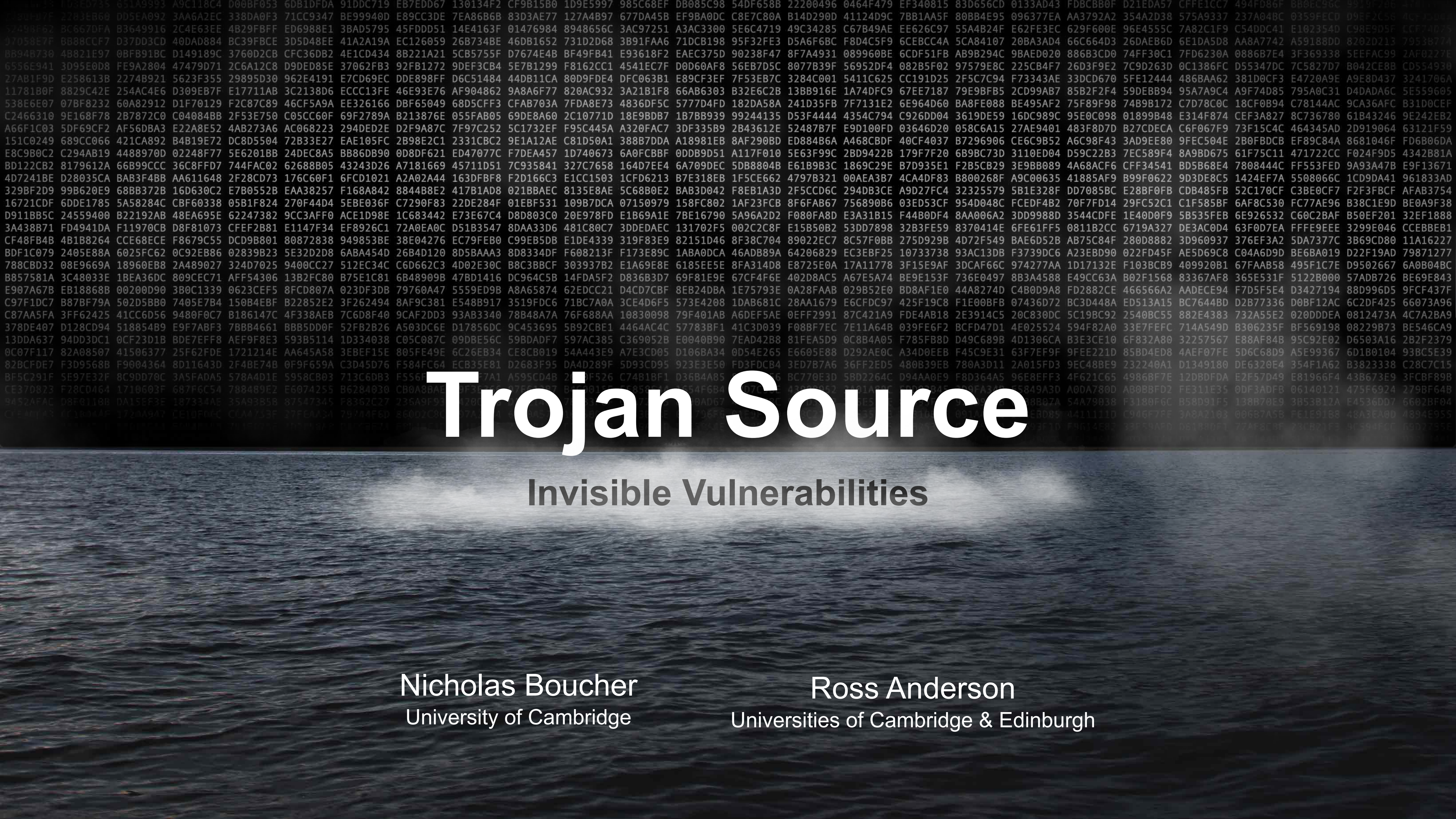


Trojan Source

Invisible Vulnerabilities

Nicholas Boucher
University of Cambridge

Ross Anderson
Universities of Cambridge & Edinburgh

Some Vulnerabilities are Invisible

UNICODE

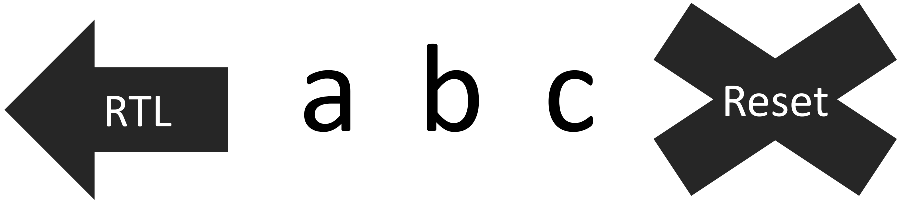
Encoding

a b c

Rendering

a b c

Encoding

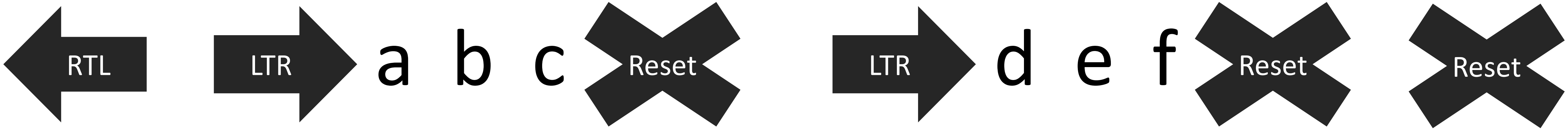


RTL a b c Reset

Rendering

c b a

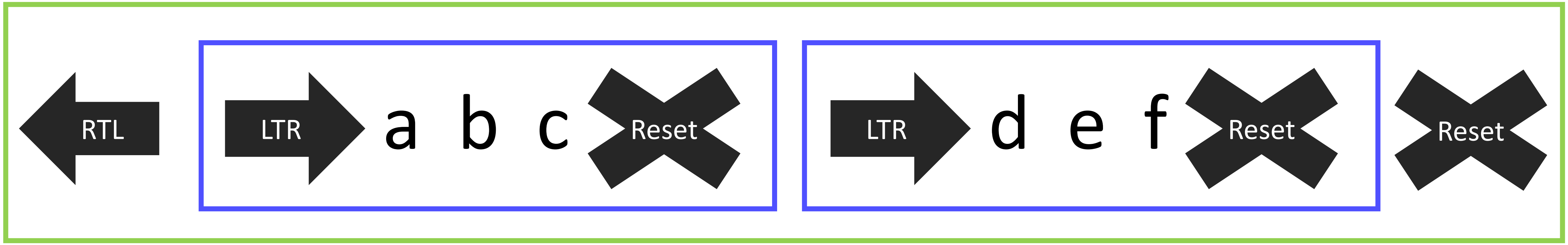
Encoding



Rendering

d e f a b c

Encoding



RTL LTR a b c Reset LTR d e f Reset Reset

Rendering

d e f a b c

Source Code?

Comments + String Literals

The Vulnerability

1. **Control characters** can override text direction.

2. This can **modify display order.**

3. They can be **placed into comments and strings.**

⇒ Evil program A to be anagrammed into benign program B.

The Vulnerability

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /* begin admins only */ if (isAdmin) {
        printf("You are an admin.\n");
    /* end admins only */ }
    return 0;
}
```

```
$> |
```

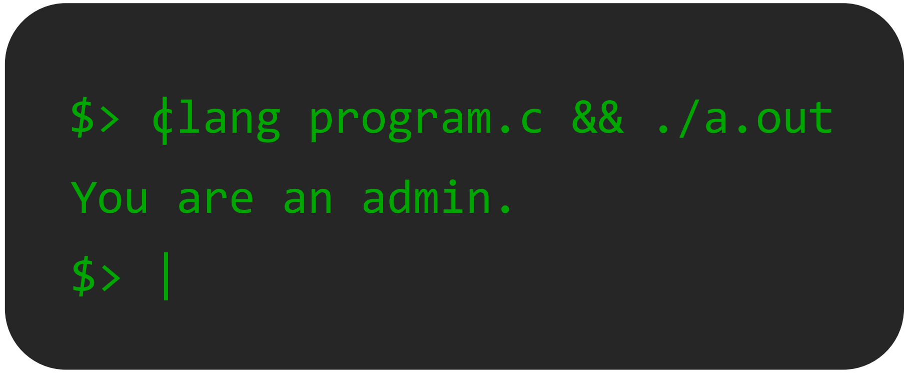
The Vulnerability

Compiler Sees
Adversarial Logic

```
/*  if (isAdmin) {  begin admins only */
```

The Vulnerability

```c
#include <stdio.h>
#include <stdbool.h>

int main() {
   bool isAdmin = false;
   /* begin admins only */ if (isAdmin) {
      printf("You are an admin.\n");
   /* end admins only */ }
   return 0;
}
```



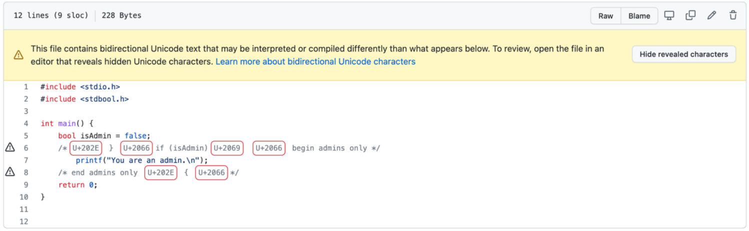
```
$> clang program.c && ./a.out
You are an admin.
$> |
```

C
JS
SQL
GO
C#
C++
BASH
$_
Java™
ASM
SOLIDITY
R
python

Supply Chain Attack

Coordinated Disclosure

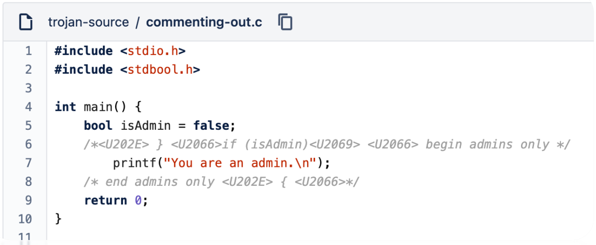
GitHub



This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters

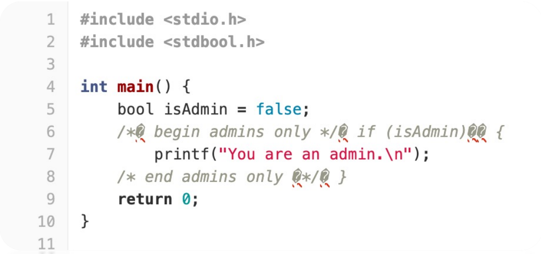
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*U+202E } U+2066 if (isAdmin) U+2069  U+2066  begin admins only */
        printf("You are an admin.\n");
    /* end admins only U+202E { U+2066 */
    return 0;
}

```

Bitbucket



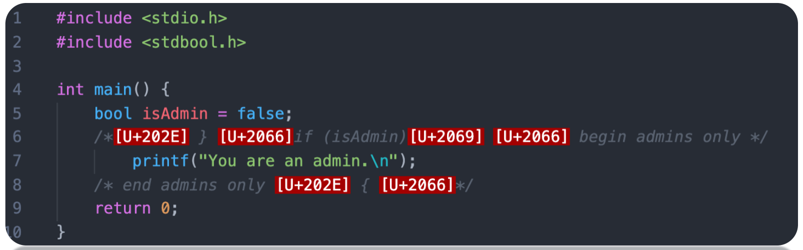
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*<U202E> } <U2066>if (isAdmin)<U2069> <U2066> begin admins only */
        printf("You are an admin.\n");
    /* end admins only <U202E> { <U2066>*/
    return 0;
}

```

GitLab



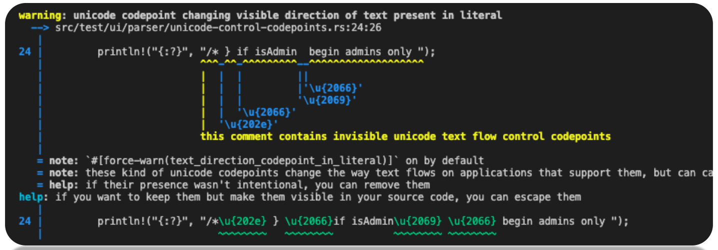
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*� begin admins only */� if (isAdmin)�� {
        printf("You are an admin.\n");
    /* end admins only �*/� }
    return 0;
}

```

VS Code



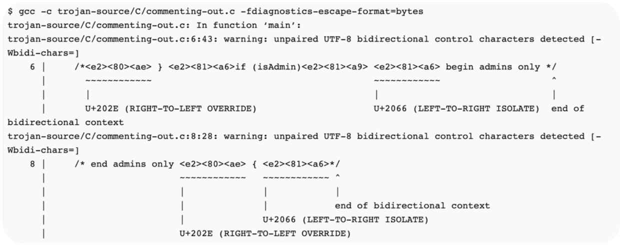
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*[U+202E] } [U+2066]if (isAdmin)[U+2069] [U+2066] begin admins only */
        printf("You are an admin.\n");
    /* end admins only [U+202E] { [U+2066]*/
    return 0;
}
```

Rust



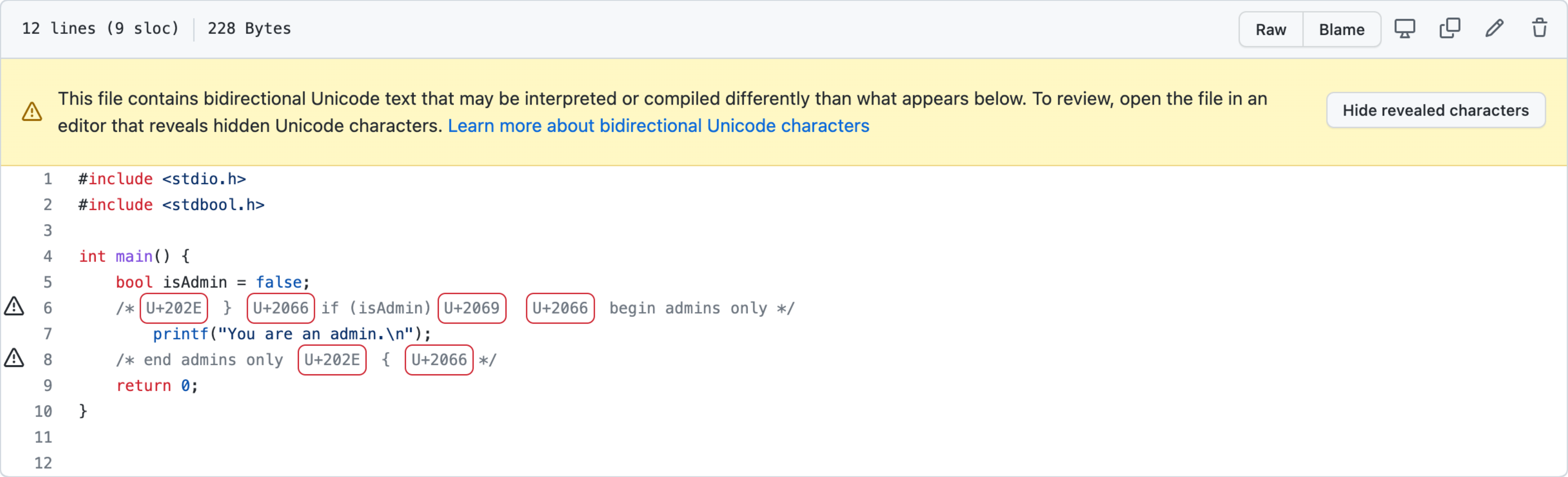
```
warning: unicode codepoint changing visible direction of text present in literal
  --> src/test/ui/parser/unicode-control-codepoints.rs:24:26
   |
24 |         println!("{:?}", "/* } if isAdmin  begin admins only ");
   |                          ^^^_^^_^^^^^^^^^^__^^^^^^^^^^^^^^^^^^^
   |                          |  |  |         ||
   |                          |  |  |         |'\u{2066}'
   |                          |  |  |         '\u{2069}'
   |                          |  |  '\u{2066}'
   |                          |  '\u{202e}'
   |                          this comment contains invisible unicode text flow control codepoints
   |
   = note: `#[force-warn(text_direction_codepoint_in_literal)]` on by default
   = note: these kind of unicode codepoints change the way text flows on applications that support them, but can ca
   = help: if their presence wasn't intentional, you can remove them
help: if you want to keep them but make them visible in your source code, you can escape them
   |
24 |         println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");
   |                             ~~~~~~~~   ~~~~~~~~          ~~~~~~~~ ~~~~~~~~
```

GCC



```
$ gcc -c trojan-source/C/commenting-out.c -fdiagnostics-escape-format=bytes
trojan-source/C/commenting-out.c: In function ‘main’:
trojan-source/C/commenting-out.c:6:43: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    6 |     /*<e2><80><ae> } <e2><81><a6>if (isAdmin)<e2><81><a9> <e2><81><a6> begin admins only */
      |       ~~~~~~~~~~~~                                         ~~~~~~~~~~~~                    ^
      |       |                                                    |                               |
      |       U+202E (RIGHT-TO-LEFT OVERRIDE)                      U+2066 (LEFT-TO-RIGHT ISOLATE)  end of
bidirectional context
trojan-source/C/commenting-out.c:8:28: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    8 |     /* end admins only <e2><80><ae> { <e2><81><a6>*/
      |                        ~~~~~~~~~~~~   ~~~~~~~~~~~~ ^
      |                        |              |            |
      |                        |              |            end of bidirectional context
      |                        |              U+2066 (LEFT-TO-RIGHT ISOLATE)
      |                        U+202E (RIGHT-TO-LEFT OVERRIDE)
```

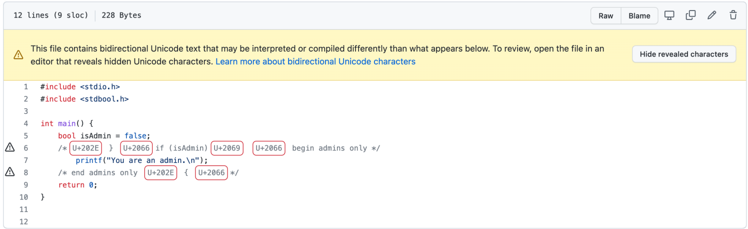
GitHub



This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*U+202E } U+2066 if (isAdmin) U+2069  U+2066 begin admins only */
        printf("You are an admin.\n");
    /* end admins only U+202E { U+2066 */
    return 0;
}


```

GitHub

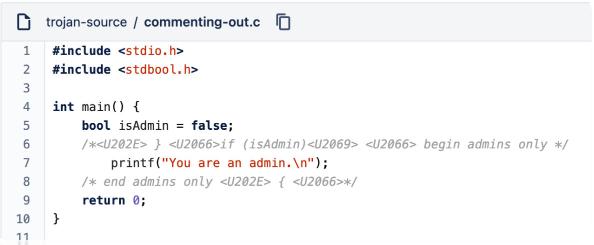


12 lines (9 sloc) | 228 Bytes

⚠ This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*U+202E } U+2066 if (isAdmin) U+2069  U+2066  begin admins only */
        printf("You are an admin.\n");
    /* end admins only U+202E { U+2066 */
    return 0;
}


```

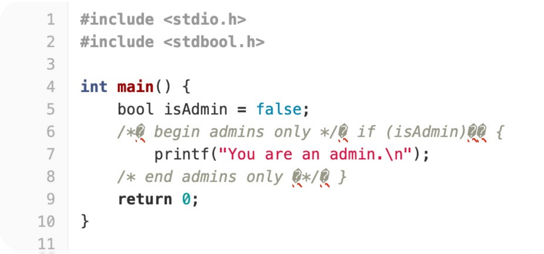
Bitbucket



trojan-source / commenting-out.c

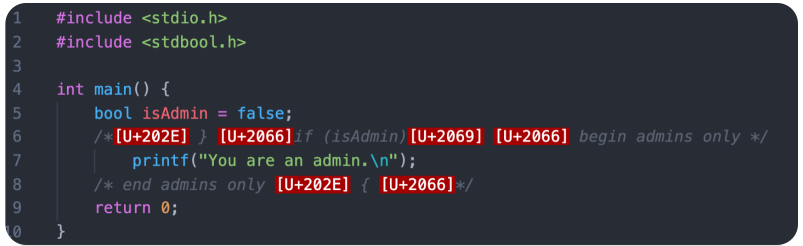
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*<U202E> } <U2066>if (isAdmin)<U2069> <U2066> begin admins only */
        printf("You are an admin.\n");
    /* end admins only <U202E> { <U2066>*/
    return 0;
}

```

GitLab



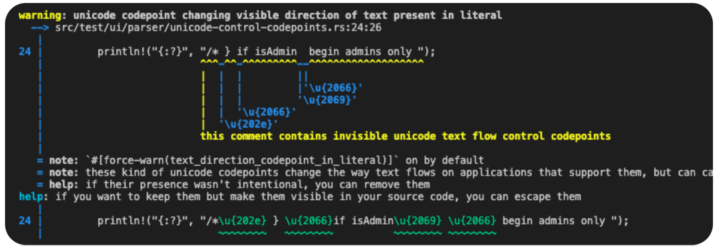
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*� begin admins only */� if (isAdmin)�� {
        printf("You are an admin.\n");
    /* end admins only �*/� }
    return 0;
}

```

VS Code



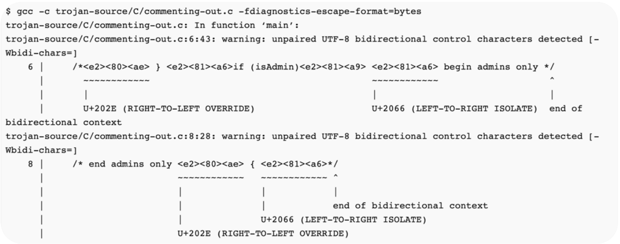
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*[U+202E] } [U+2066]if (isAdmin)[U+2069] [U+2066] begin admins only */
        printf("You are an admin.\n");
    /* end admins only [U+202E] { [U+2066]*/
    return 0;
}
```

Rust



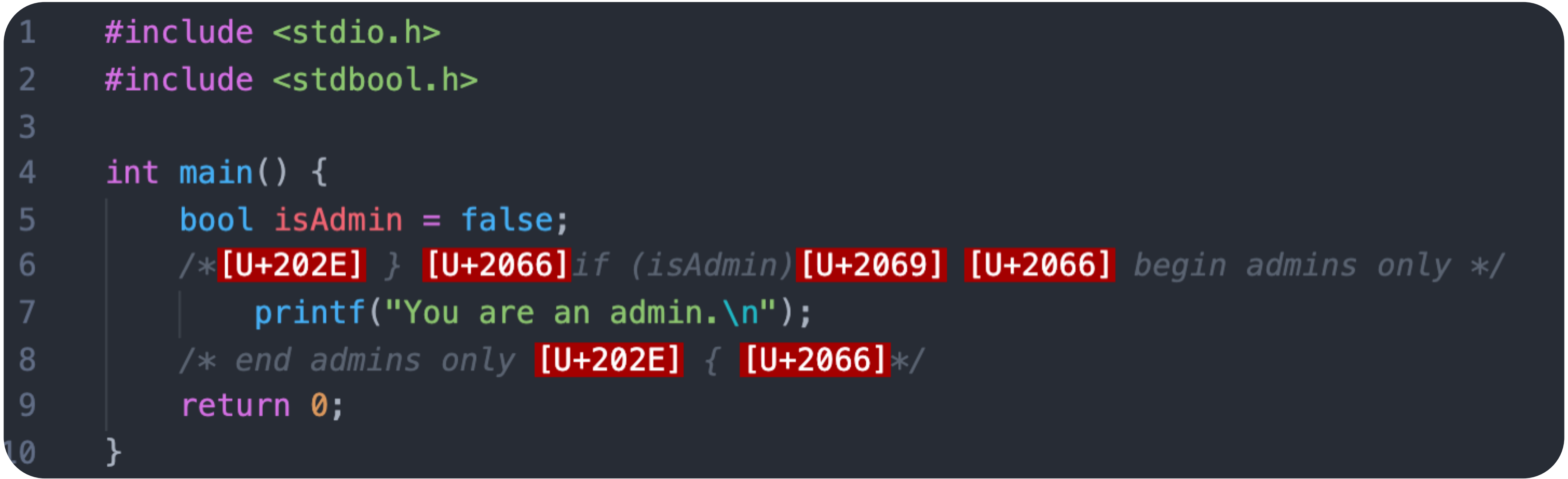
```
warning: unicode codepoint changing visible direction of text present in literal
  --> src/test/ui/parser/unicode-control-codepoints.rs:24:26
   |
24 |         println!("{:?}", "/* } if isAdmin  begin admins only ");
   |                           ^^^_^^_^^^^^^^^^__^^^^^^^^^^^^^^^^^^^
   |                           |  |  |         ||
   |                           |  |  |         |'\u{2066}'
   |                           |  |  |         '\u{2069}'
   |                           |  |  '\u{2066}'
   |                           |  '\u{202e}'
   |                           this comment contains invisible unicode text flow control codepoints
   |
   = note: `#[force-warn(text_direction_codepoint_in_literal)]` on by default
   = note: these kind of unicode codepoints change the way text flows on applications that support them, but can ca
   = help: if their presence wasn't intentional, you can remove them
help: if you want to keep them but make them visible in your source code, you can escape them
   |
24 |         println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");
   |                             ~~~~~~~~   ~~~~~~~~         ~~~~~~~~ ~~~~~~~~
```

GCC



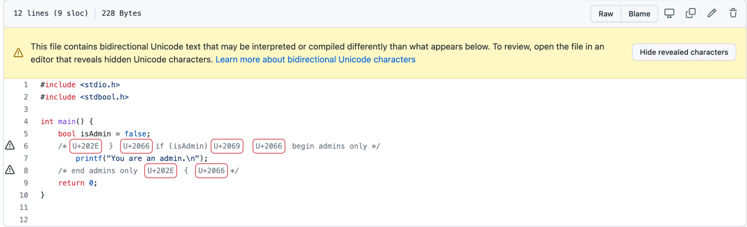
```
$ gcc -c trojan-source/C/commenting-out.c -fdiagnostics-escape-format=bytes
trojan-source/C/commenting-out.c: In function ‘main’:
trojan-source/C/commenting-out.c:6:43: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    6 |     /*<e2><80><ae> } <e2><81><a6>if (isAdmin)<e2><81><a9> <e2><81><a6> begin admins only */
      |       ~~~~~~~~~~~~                                        ~~~~~~~~~~~~                ^
      |       |                                                   |                           |
      |       U+202E (RIGHT-TO-LEFT OVERRIDE)                     U+2066 (LEFT-TO-RIGHT ISOLATE)  end of
bidirectional context
trojan-source/C/commenting-out.c:8:28: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    8 |     /* end admins only <e2><80><ae> { <e2><81><a6>*/
      |                        ~~~~~~~~~~~~   ~~~~~~~~~~~~ ^
      |                        |              |            |
      |                        |              |            end of bidirectional context
      |                        |              U+2066 (LEFT-TO-RIGHT ISOLATE)
      |                        U+202E (RIGHT-TO-LEFT OVERRIDE)
```

VS Code



```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*[U+202E] } [U+2066]if (isAdmin)[U+2069] [U+2066] begin admins only */
        printf("You are an admin.\n");
    /* end admins only [U+202E] { [U+2066]*/
    return 0;
}
```

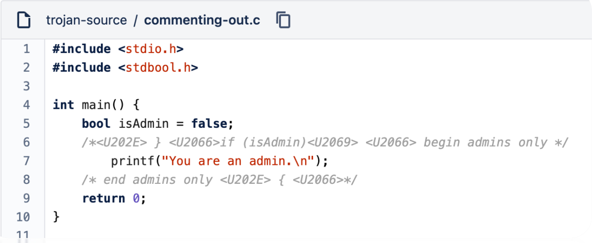
GitHub



This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters

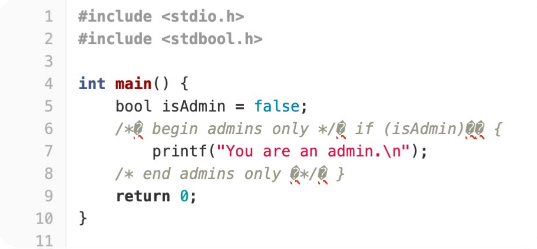
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*U+202E } U+2066 if (isAdmin) U+2069  U+2066  begin admins only */
        printf("You are an admin.\n");
    /* end admins only U+202E  {  U+2066 */
    return 0;
}


```

Bitbucket



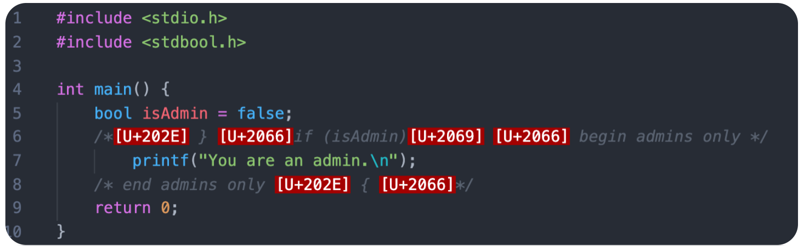
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*<U202E> } <U2066>if (isAdmin)<U2069> <U2066> begin admins only */
        printf("You are an admin.\n");
    /* end admins only <U202E> { <U2066>*/
    return 0;
}

```

GitLab



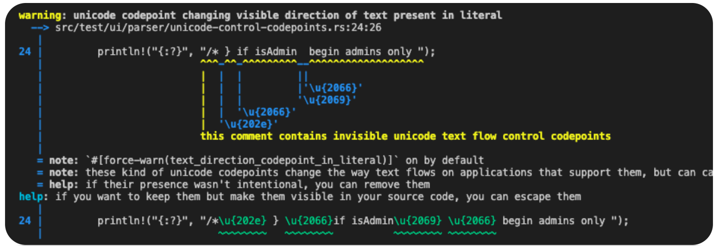
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*� begin admins only */� if (isAdmin)�� {
        printf("You are an admin.\n");
    /* end admins only �*/� }
    return 0;
}

```

VS Code



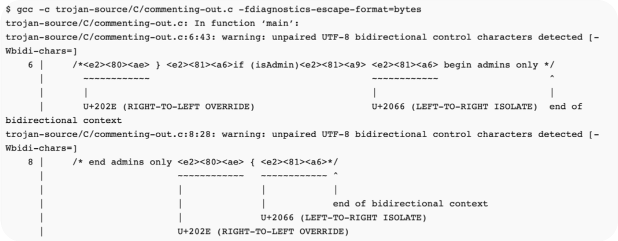
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*[U+202E] } [U+2066]if (isAdmin)[U+2069] [U+2066] begin admins only */
        printf("You are an admin.\n");
    /* end admins only [U+202E] { [U+2066]*/
    return 0;
}
```

Rust



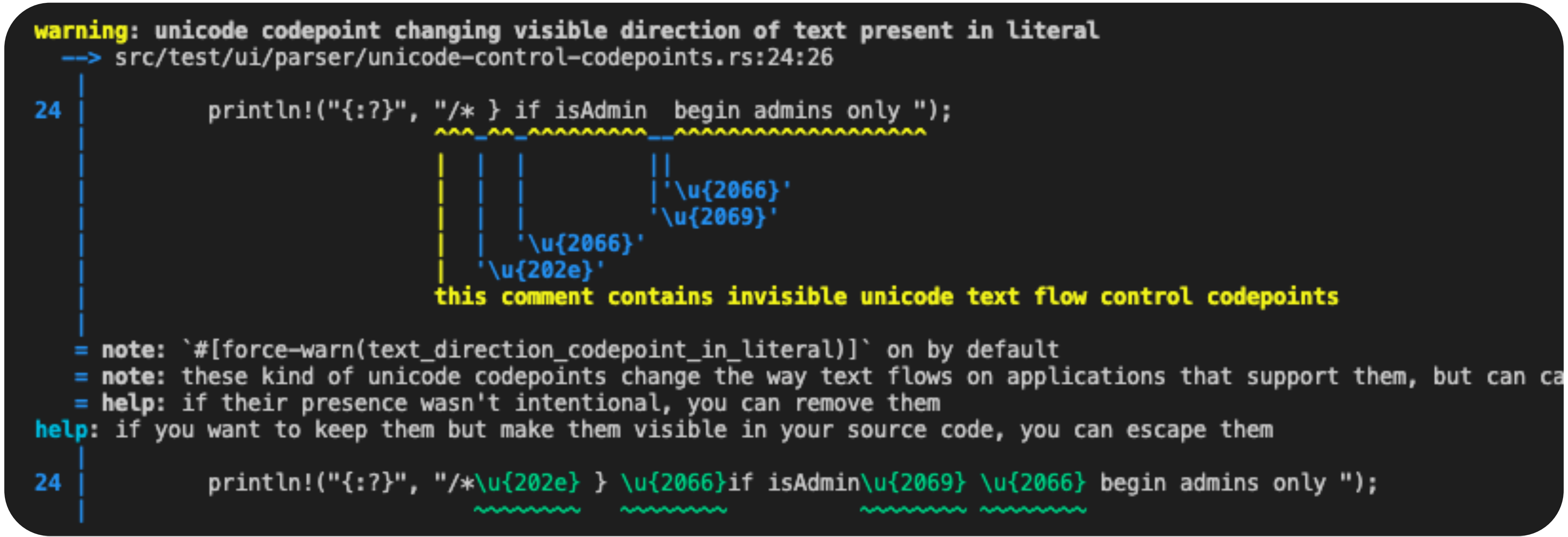
```
warning: unicode codepoint changing visible direction of text present in literal
  --> src/test/ui/parser/unicode-control-codepoints.rs:24:26
   |
24 |         println!("{:?}", "/* } if isAdmin  begin admins only ");
   |                           ^^^_^^_^^^^^^^^^__^^^^^^^^^^^^^^^^^^^^
   |                           |  |  |         ||
   |                           |  |  |         |'\u{2066}'
   |                           |  |  |         '\u{2069}'
   |                           |  |  '\u{2066}'
   |                           |  '\u{202e}'
   |                           this comment contains invisible unicode text flow control codepoints
   |
   = note: `#[force-warn(text_direction_codepoint_in_literal)]` on by default
   = note: these kind of unicode codepoints change the way text flows on applications that support them, but can ca
   = help: if their presence wasn't intentional, you can remove them
help: if you want to keep them but make them visible in your source code, you can escape them
   |
24 |         println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");
   |                             ~~~~~~~~   ~~~~~~~~          ~~~~~~~~ ~~~~~~~~
```

GCC



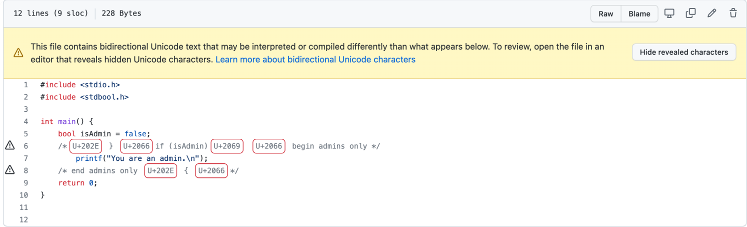
```
$ gcc -c trojan-source/C/commenting-out.c -fdiagnostics-escape-format=bytes
trojan-source/C/commenting-out.c: In function ‘main’:
trojan-source/C/commenting-out.c:6:43: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    6 |     /*<e2><80><ae> } <e2><81><a6>if (isAdmin)<e2><81><a9> <e2><81><a6> begin admins only */
      |       ~~~~~~~~~~~~                                        ~~~~~~~~~~~~                     ^
      |       |                                                   |                                |
      |       U+202E (RIGHT-TO-LEFT OVERRIDE)                     U+2066 (LEFT-TO-RIGHT ISOLATE)  end of
bidirectional context
trojan-source/C/commenting-out.c:8:28: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    8 |     /* end admins only <e2><80><ae> { <e2><81><a6>*/
      |                        ~~~~~~~~~~~~   ~~~~~~~~~~~~ ^
      |                        |              |            |
      |                        |              |            end of bidirectional context
      |                        |              U+2066 (LEFT-TO-RIGHT ISOLATE)
      |                        U+202E (RIGHT-TO-LEFT OVERRIDE)
```

Rust



```
warning: unicode codepoint changing visible direction of text present in literal
  --> src/test/ui/parser/unicode-control-codepoints.rs:24:26
   |
24 |         println!("{:?}", "/* } if isAdmin  begin admins only ");
   |                          ^^^_^^_^^^^^^^^^__^^^^^^^^^^^^^^^^^^^
   |                          |  |  |         ||
   |                          |  |  |         |'\u{2066}'
   |                          |  |  |         '\u{2069}'
   |                          |  |  '\u{2066}'
   |                          |  '\u{202e}'
   |                          this comment contains invisible unicode text flow control codepoints
   |
   = note: `#[force-warn(text_direction_codepoint_in_literal)]` on by default
   = note: these kind of unicode codepoints change the way text flows on applications that support them, but can ca
   = help: if their presence wasn't intentional, you can remove them
help: if you want to keep them but make them visible in your source code, you can escape them
   |
24 |         println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");
   |                             ~~~~~~~~   ~~~~~~~~          ~~~~~~~~ ~~~~~~~~
```

GitHub

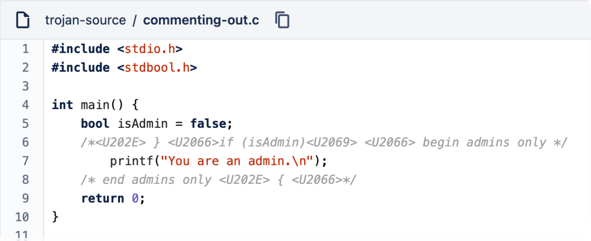


```
12 lines (9 sloc) | 228 Bytes
```

⚠ This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /* U+202E } U+2066 if (isAdmin) U+2069  U+2066 begin admins only */
        printf("You are an admin.\n");
    /* end admins only U+202E { U+2066 */
    return 0;
}


```

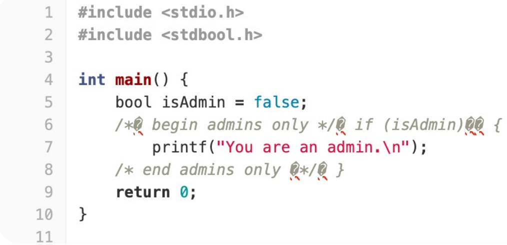
Bitbucket



trojan-source / commenting-out.c

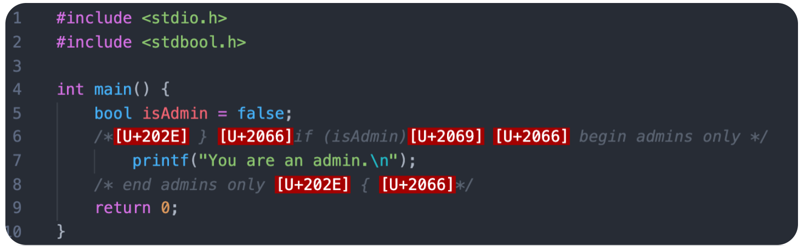
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*<U202E> } <U2066>if (isAdmin)<U2069> <U2066> begin admins only */
        printf("You are an admin.\n");
    /* end admins only <U202E> { <U2066>*/
    return 0;
}

```

GitLab



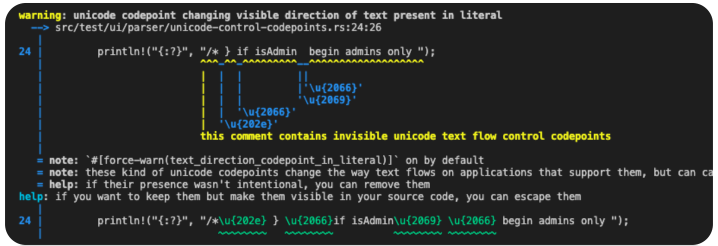
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*@ begin admins only */@ if (isAdmin)@@ {
        printf("You are an admin.\n");
    /* end admins only @*/@ }
    return 0;
}

```

VS Code



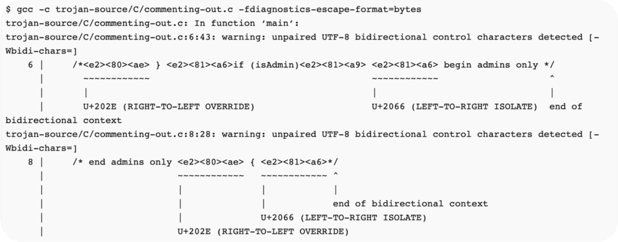
```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /*[U+202E] } [U+2066]if (isAdmin)[U+2069] [U+2066] begin admins only */
        printf("You are an admin.\n");
    /* end admins only [U+202E] { [U+2066]*/
    return 0;
}
```

Rust

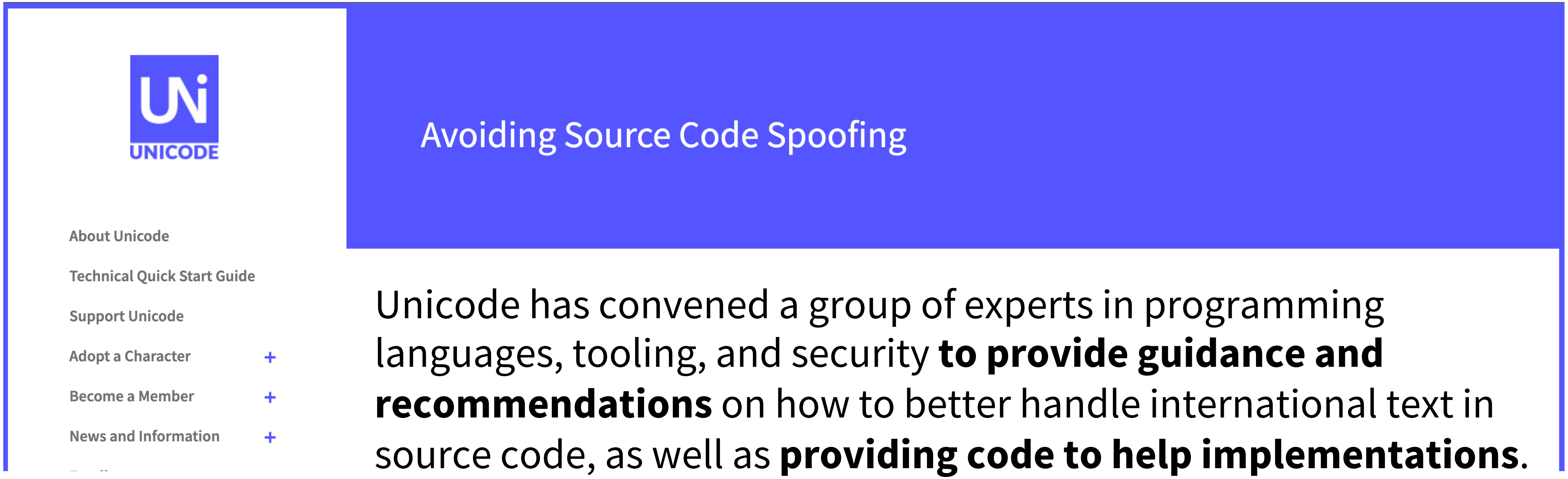


```
warning: unicode codepoint changing visible direction of text present in literal
  --> src/test/ui/parser/unicode-control-codepoints.rs:24:26
   |
24 |         println!("{:?}", "/* } if isAdmin  begin admins only ");
   |                          ^^^_^^_^^^^^^^^^__^^^^^^^^^^^^^^^^^^^
   |                          |  |  |         ||
   |                          |  |  |         |'\u{2066}'
   |                          |  |  |         '\u{2069}'
   |                          |  |  '\u{2066}'
   |                          |  '\u{202e}'
   |                          this comment contains invisible unicode text flow control codepoints
   |
   = note: `#[force-warn(text_direction_codepoint_in_literal)]` on by default
   = note: these kind of unicode codepoints change the way text flows on applications that support them, but can ca
   = help: if their presence wasn't intentional, you can remove them
help: if you want to keep them but make them visible in your source code, you can escape them
   |
24 |         println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");
   |                            ~~~~~~~~   ~~~~~~~~          ~~~~~~~~ ~~~~~~~~
```

GCC



```
$ gcc -c trojan-source/C/commenting-out.c -fdiagnostics-escape-format=bytes
trojan-source/C/commenting-out.c: In function ‘main’:
trojan-source/C/commenting-out.c:6:43: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    6 |     /*<e2><80><ae> } <e2><81><a6>if (isAdmin)<e2><81><a9> <e2><81><a6> begin admins only */
      |       ~~~~~~~~~~~~                                        ~~~~~~~~~~~~                  ^
      |       |                                                   |                             |
      |       U+202E (RIGHT-TO-LEFT OVERRIDE)                     U+2066 (LEFT-TO-RIGHT ISOLATE)  end of
bidirectional context
trojan-source/C/commenting-out.c:8:28: warning: unpaired UTF-8 bidirectional control characters detected [-
Wbidi-chars=]
    8 |     /* end admins only <e2><80><ae> { <e2><81><a6>*/
      |                        ~~~~~~~~~~~~   ~~~~~~~~~~~~ ^
      |                        |              |            |
      |                        |              |            end of bidirectional context
      |                        |              U+2066 (LEFT-TO-RIGHT ISOLATE)
      |                        U+202E (RIGHT-TO-LEFT OVERRIDE)
```



Avoiding Source Code Spoofing

Unicode has convened a group of experts in programming languages, tooling, and security **to provide guidance and recommendations** on how to better handle international text in source code, as well as **providing code to help implementations**.

See paper for…

- Attack Generation
- Attack Detection
- Attack Variants
- CVE Details
- Compiler vs Editor vs Repo
- GitHub Ecosystem Scanning
- Coordinated Disclosure Details
- Industry Response
- Defenses

… and more

It’s not just source code

• **LLMs**	*Bad Characters: Imperceptible NLP Attacks* *(S&P 2022)*
• **Search Engines**	*Boosting Big Brother: Attacking Search Engines with Encodings* *(RAID 2023)*
• **Coordinated Disclosure**	*Talking Trojan: Analyzing an Industry-Wide Disclosure* *(SCORED 2022)*

More Info

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

https://trojansource.codes