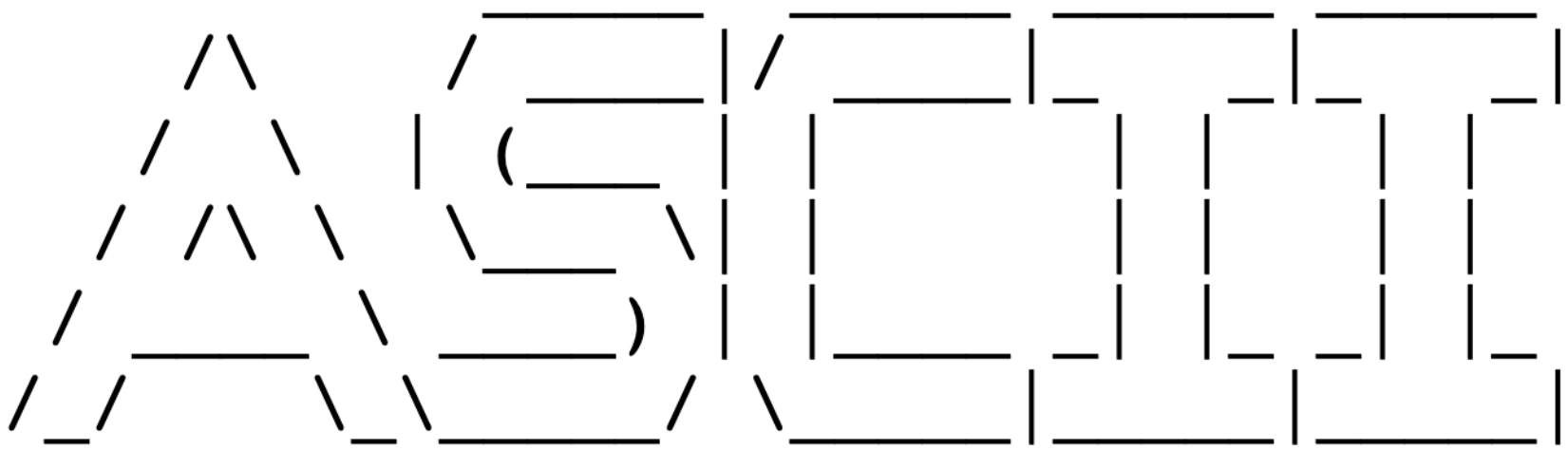
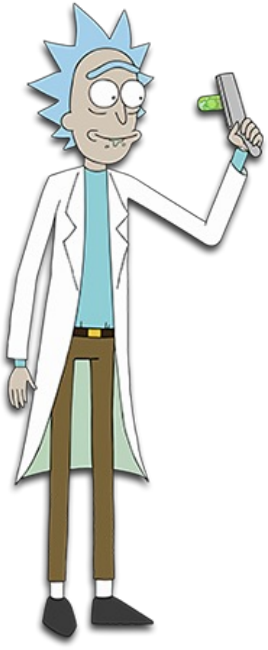
Trojan Source and Bad Characters:

Invisible Hacks and Reluctant Patching

Nicholas Boucher & Ross Anderson

Invisible Attacks


ASCII



UNICODE

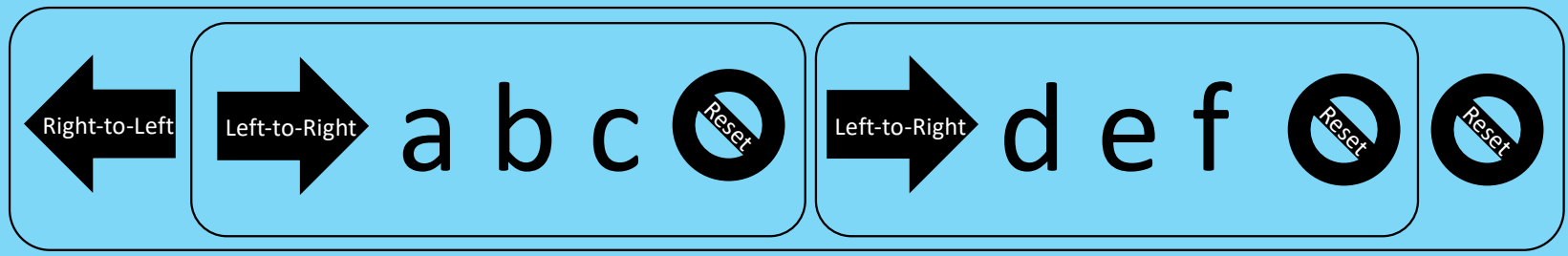
Hello, world!

مرحبا بالعالم!

Bidirectional Algorithm

Encoded Bytes

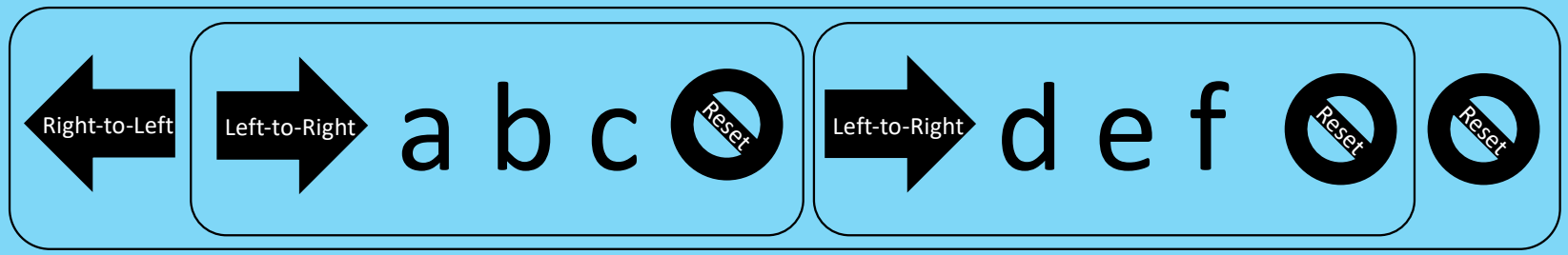
Rendered Text



Encoded Bytes

Rendered Text

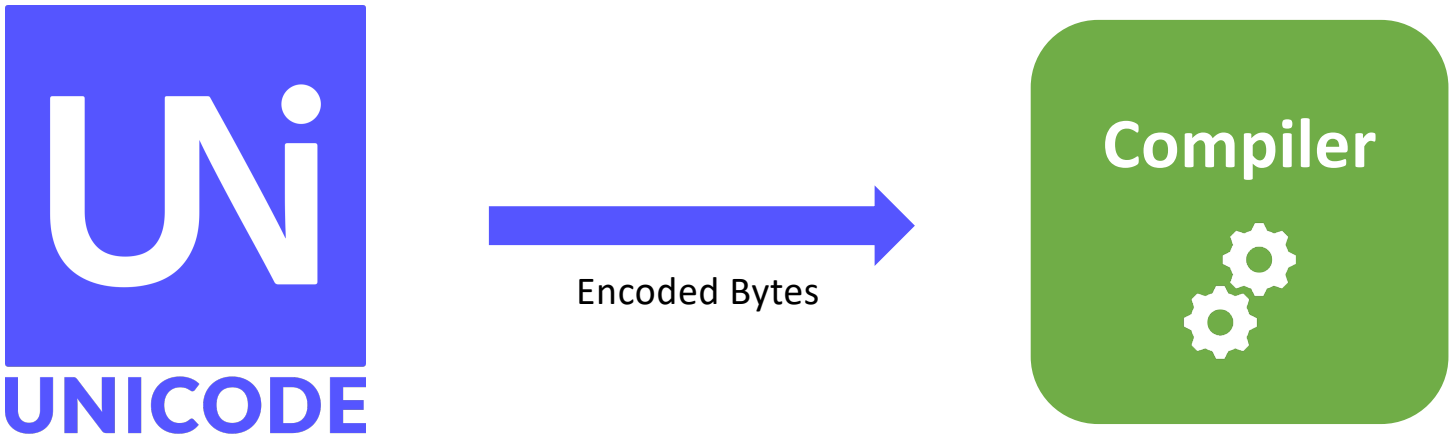
d b c a e f

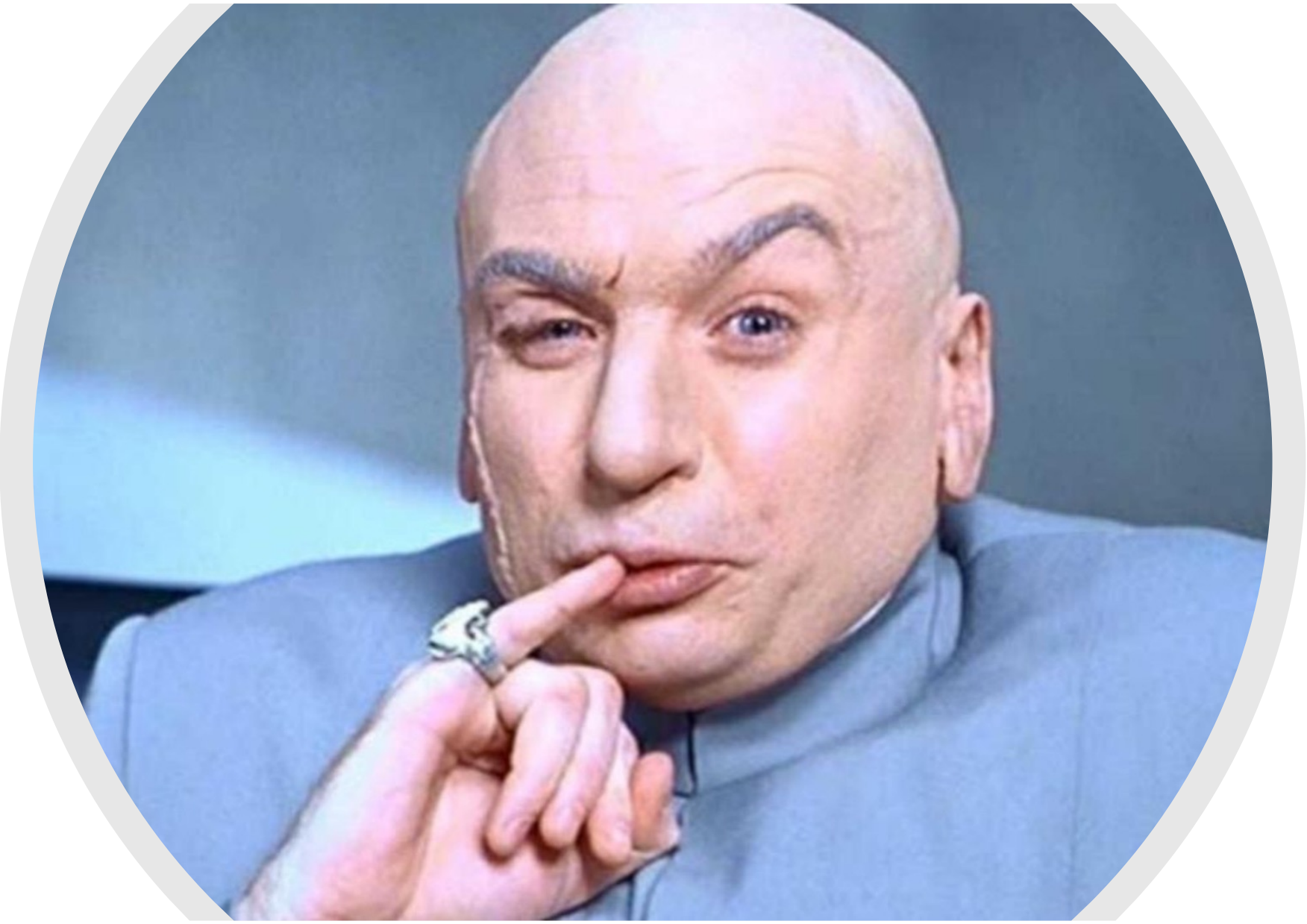


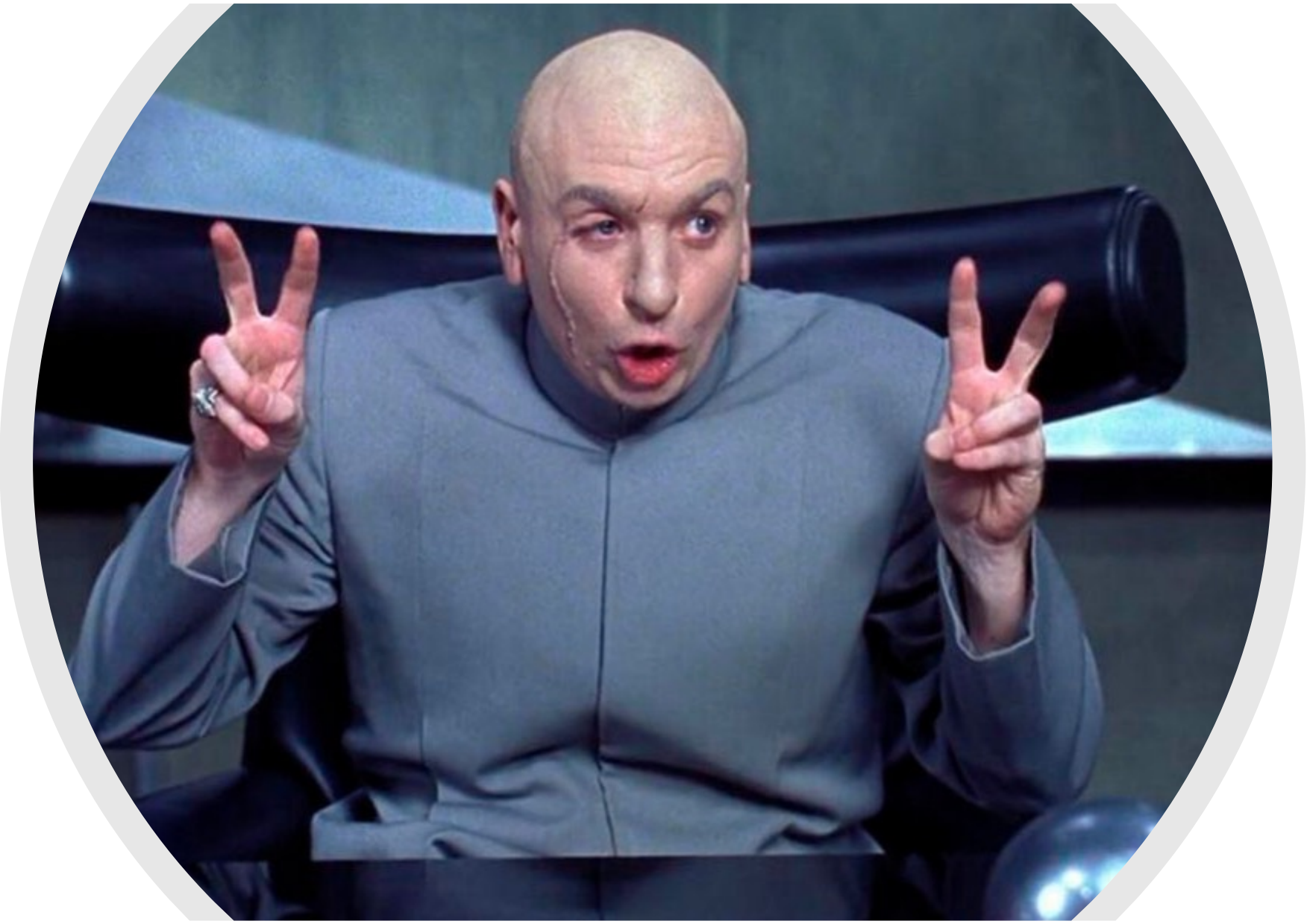
Encoded Bytes

Rendered Text

d e f a b c


UNICODE
Encoded Bytes
Compiler





Text is written both left and right

Unicode has directionality control characters

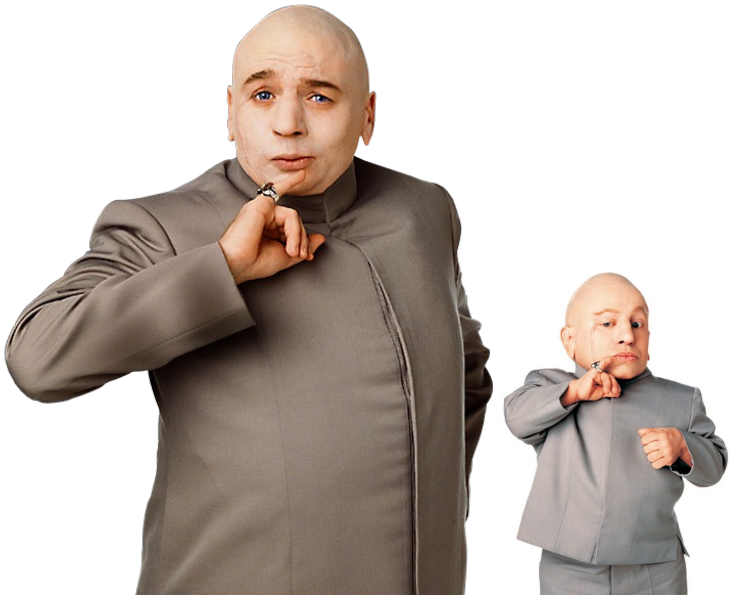
Control characters can go in comments and strings

Directionality **control characters**

in comments and strings

can **reorder source code**

to change its logic

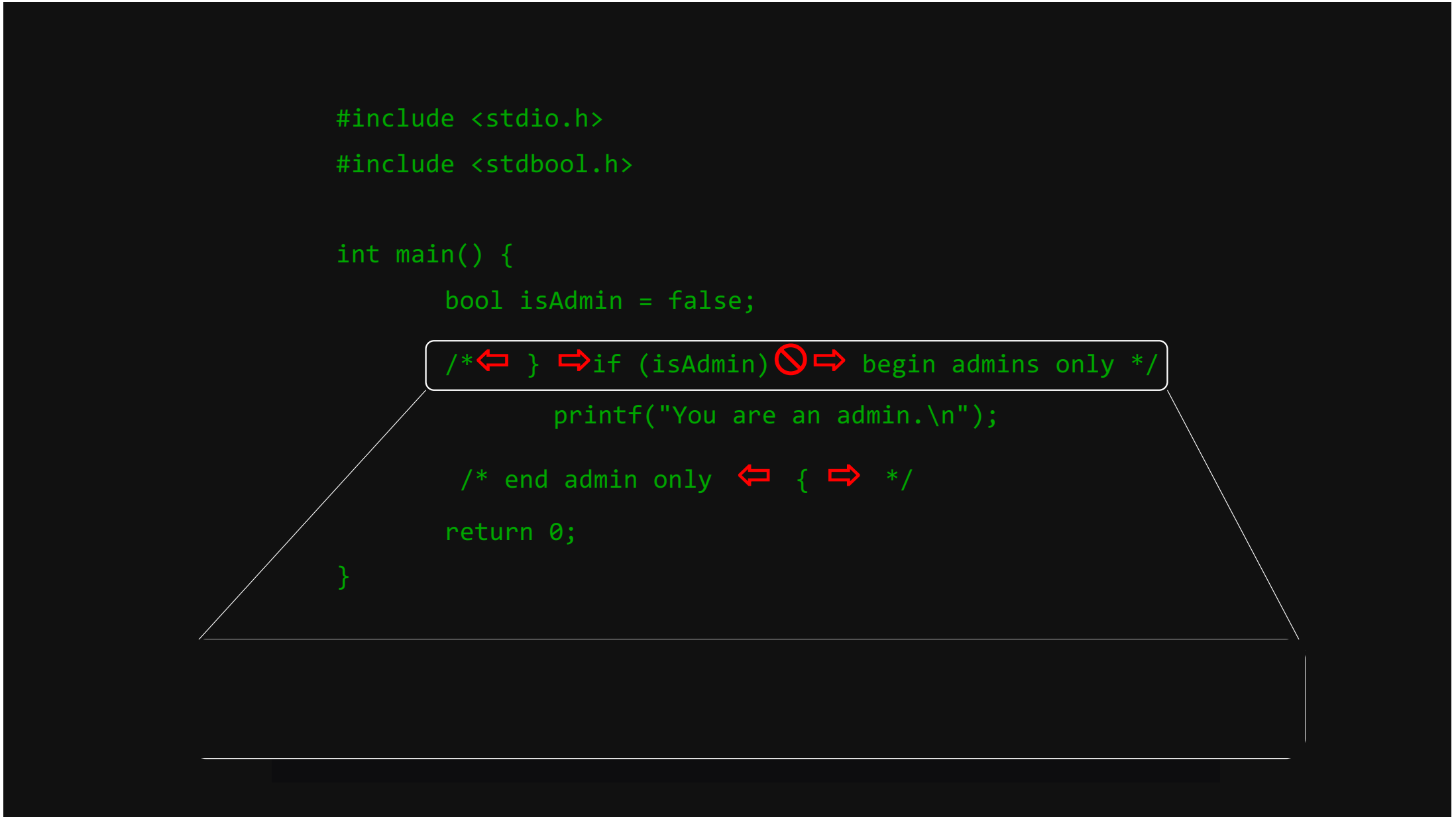


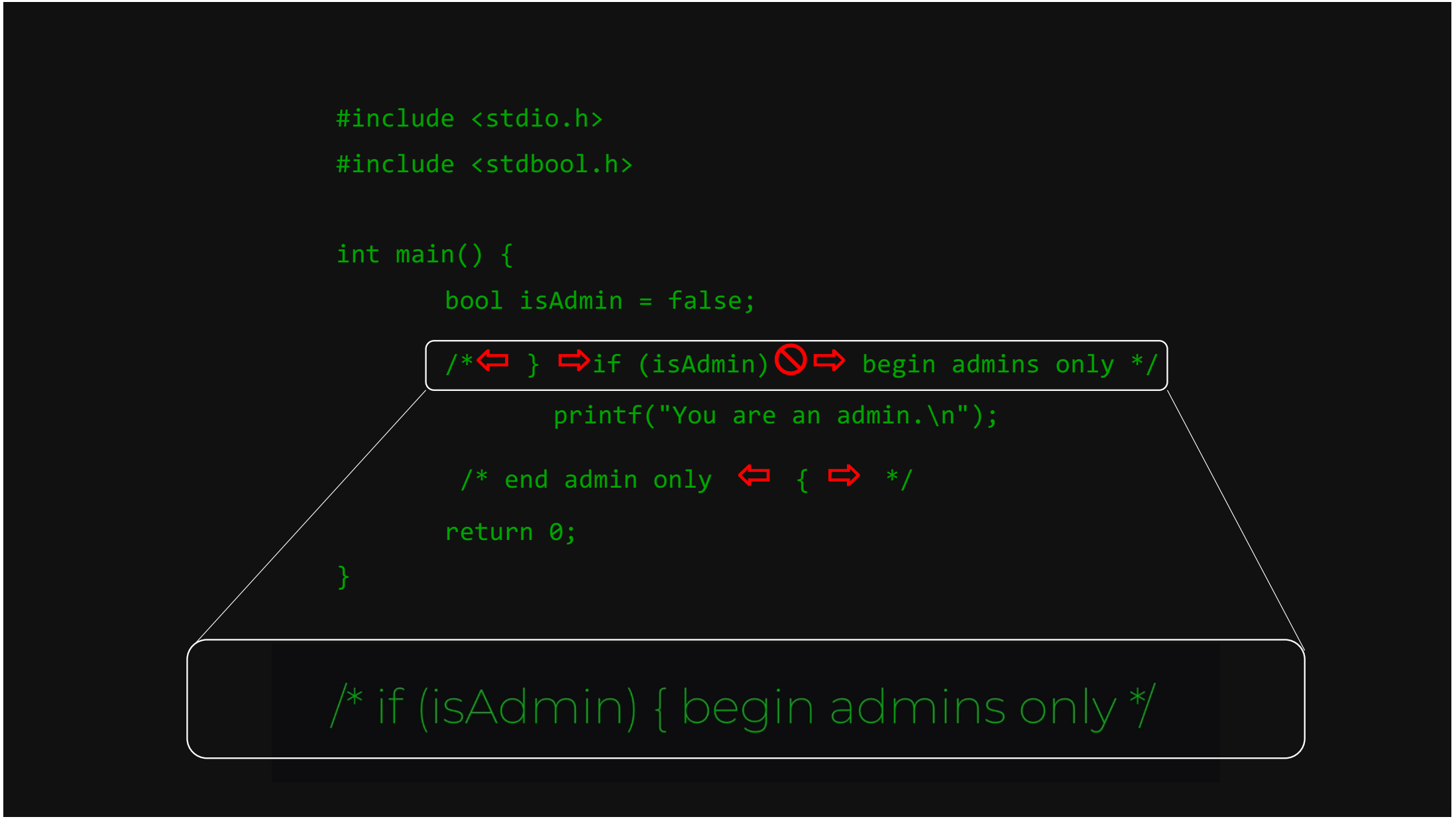
Example

Dark Mode

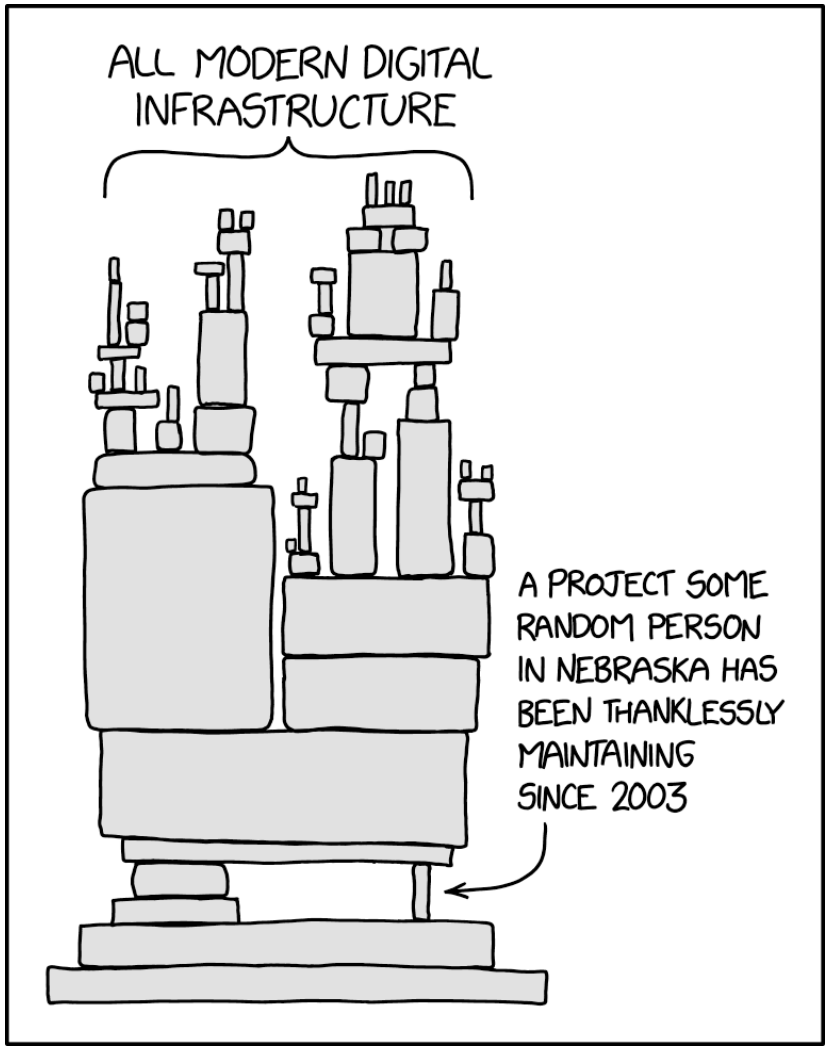
```c
#include <stdio.h>
#include <stdbool.h>

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

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


#include <stdio.h>
#include <stdbool.h>
int main() {
bool isAdmin = false;
/* } if (isAdmin) begin admins only */
printf("You are an admin.\n");
/* end admin only { */
return 0;
}


#include <stdio.h>
#include <stdbool.h>
int main() {
bool isAdmin = false;
/* ⇦ } ⇨ if (isAdmin) ⃠ ⇨ begin admins only */
printf("You are an admin.\n");
/* end admin only ⇦ { ⇨ */
return 0;
}
/* if (isAdmin) { begin admins only */

Trojan Source Attack



xkcd.com/2347

Optimizing some code #16

Open **drevil** wants to merge 1 commit into `nickboucher:main` from `drevil:main`

Conversation 0 | Commits 1 | Checks 0 | Files changed 9

ibuclaw commented on Nov 15, 2021 — First-time contributor

Definitely not invisible malware.

❤️ 1

 Add the optimizations a8fa68e

Add more commits by pushing to the **`drevil`** branch on **drevil/trojan-source**.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Optimizing some code #16

Open **drevil** wants to merge 1 commit into `nickboucher:main` from `drevil:main`

Conversation 0 | Commits 1 | Checks 0 | Files changed 9

ibuclaw commented on Nov 15, 2021 — First-time contributor

Definitely not invisible malware.

❤️ 1

 Add the optimizations a8fa68e

Add more commits by pushing to the **`drevil`** branch on **drevil/trojan-source**.

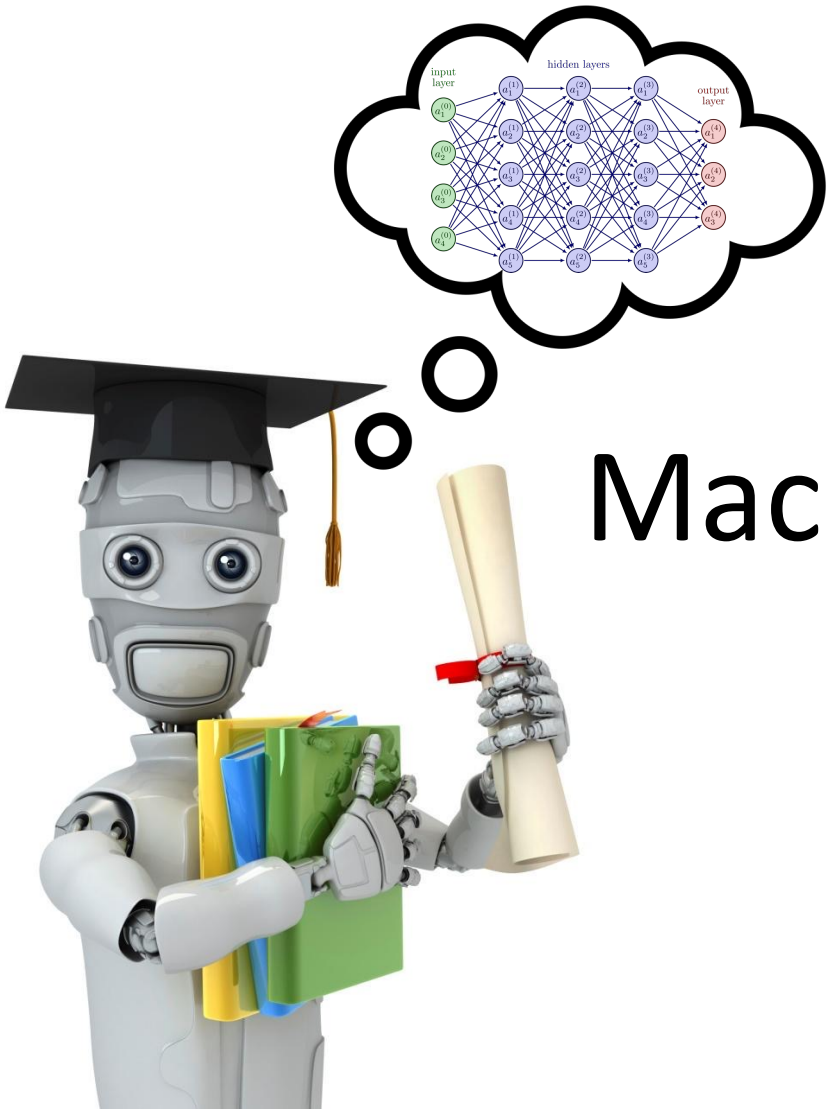
This branch has no conflicts with the base branch

Merging can be performed automatically.

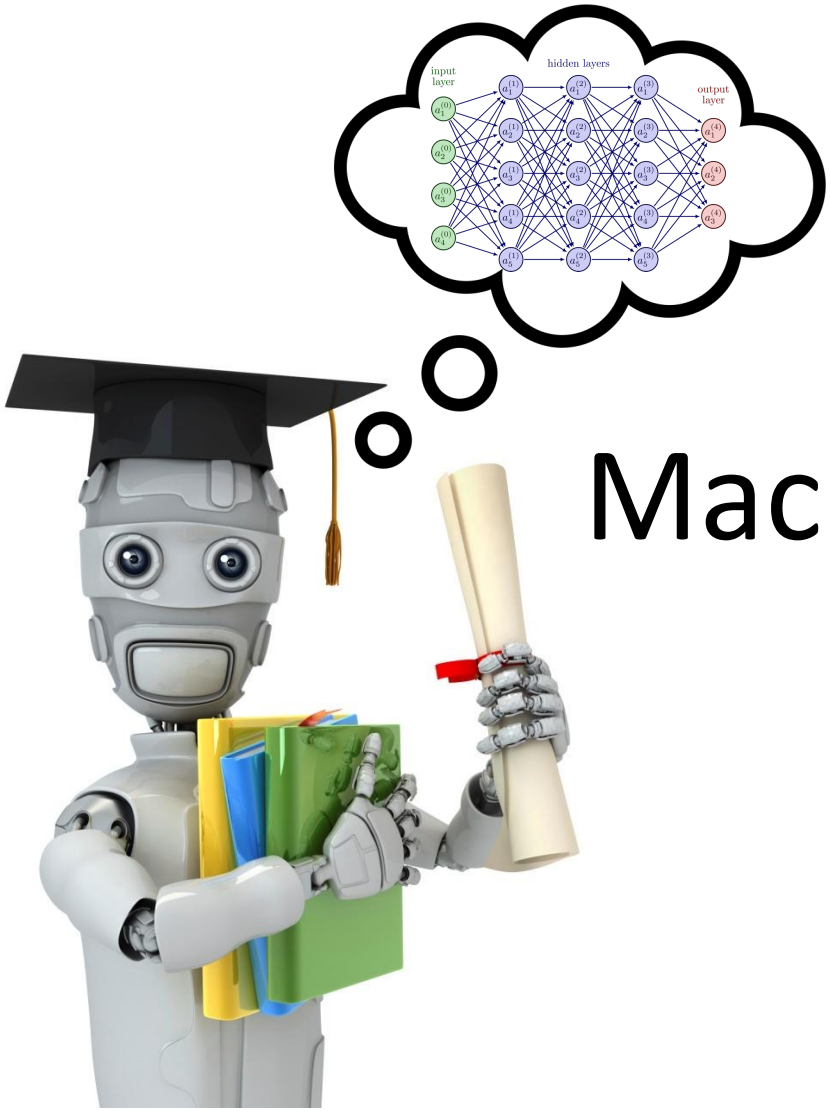
Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Trojan Source Attack

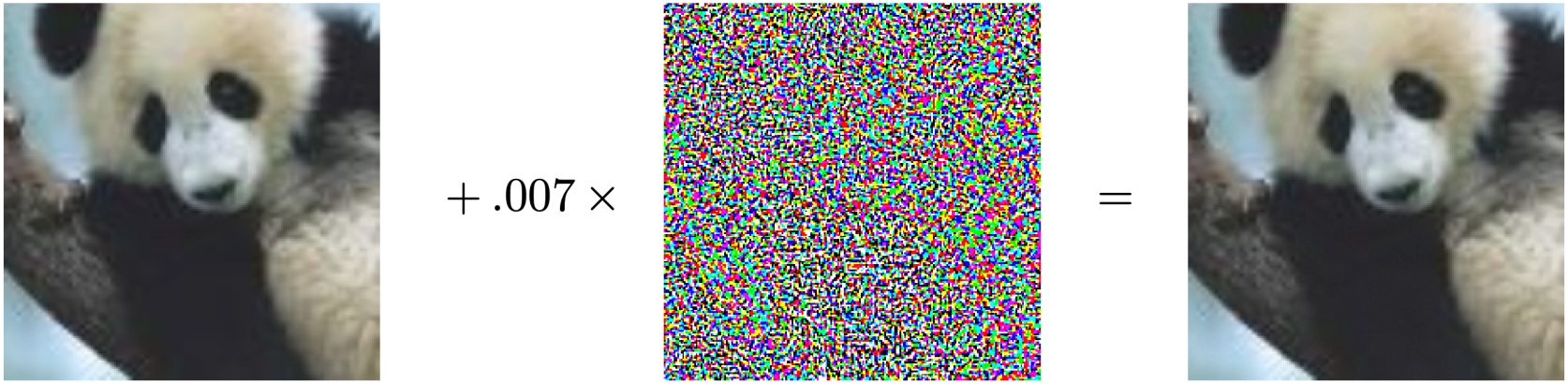


Machine Learning



Machine Learning

Adversarial Examples



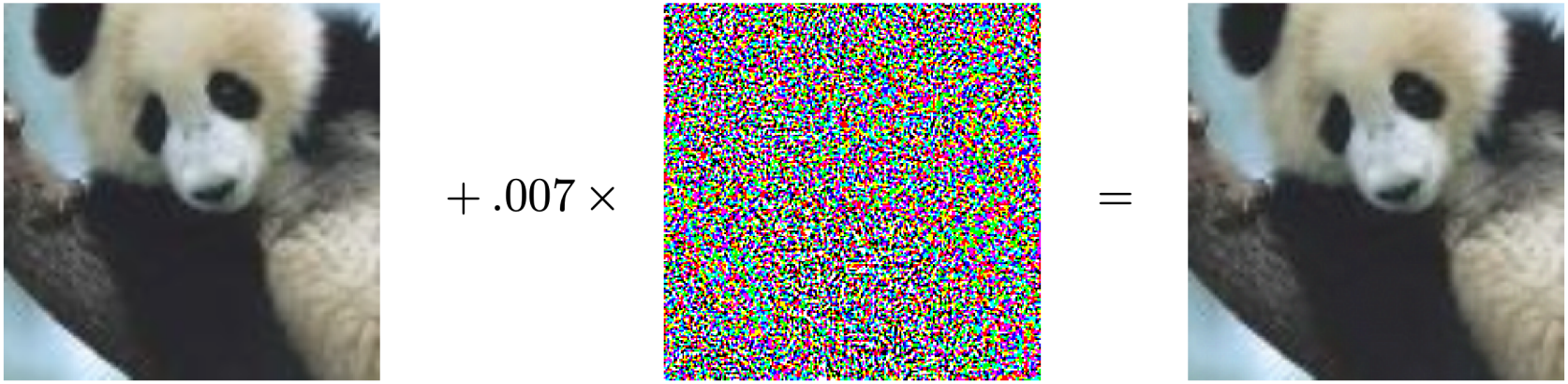
"panda"
57.7% confidence

"nematode"
8.2% confidence

"gibbon"
99.3 % confidence

I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

Adversarial Examples



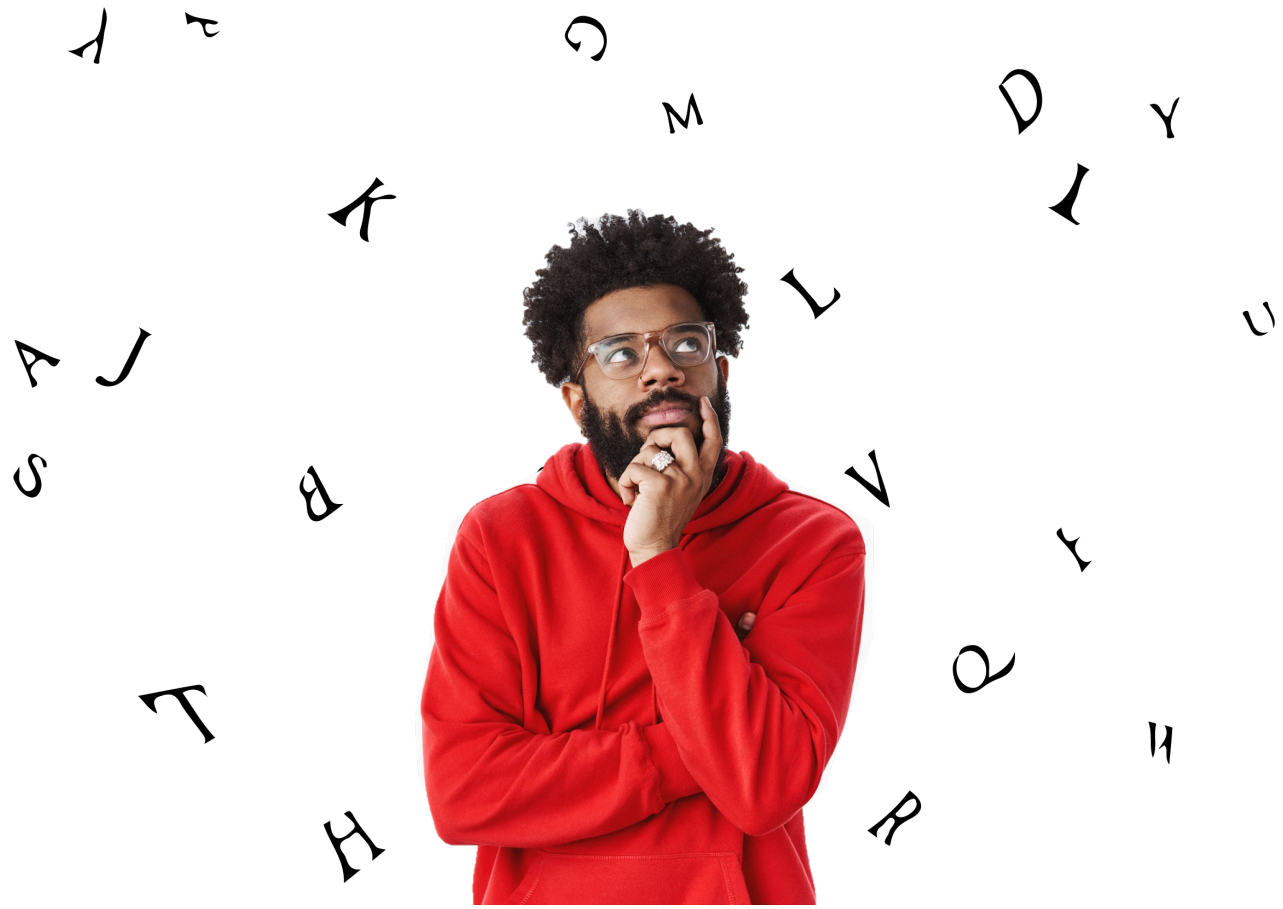
"panda"
57.7% confidence

"nematode"
8.2% confidence

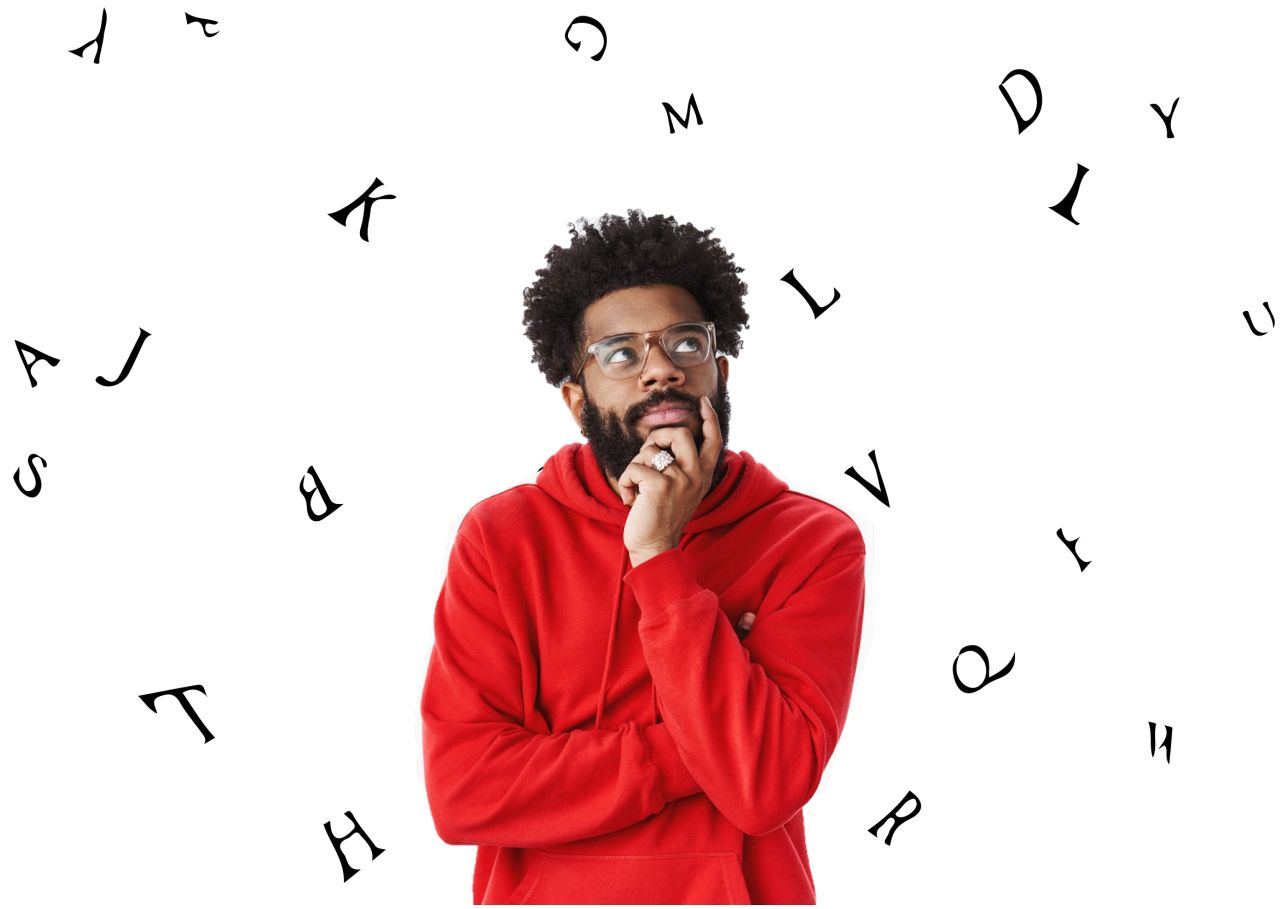
"gibbon"
99.3 % confidence

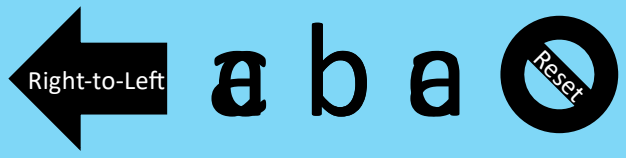
I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

Text Adversarial Examples



Text Adversarial Examples





Encoded Bytes

Rendered Text

a b c

Encoded Bytes

Rendered Text

a b c

Text Adversarial Examples

+

Bidirectional Control Characters

=

Non-Toxic Content

Text Adversarial Examples

\+ Invisible Characters = Non-Toxic Content

Example: Invisible Characters

Zero Width Space

Text Adversarial Examples

\+ Homoglyphs = Non-Toxic Content

Example: Homoglyphs

H H

Cyrillic Latin

Text Adversarial Examples

+ Deletions = Non-Toxic Content

Example: Deletions

X Y Z ⌫ ⌫ ⌫

Text Adversarial Examples

+ Unicode Tricks = Non-Toxic Content

Text Adversarial Examples

Input + Unicode Tricks = Adversarial Output

Send money to account **1234**

Envoyez de l’argent au compte **4321**

Unicode breaks machine learning.

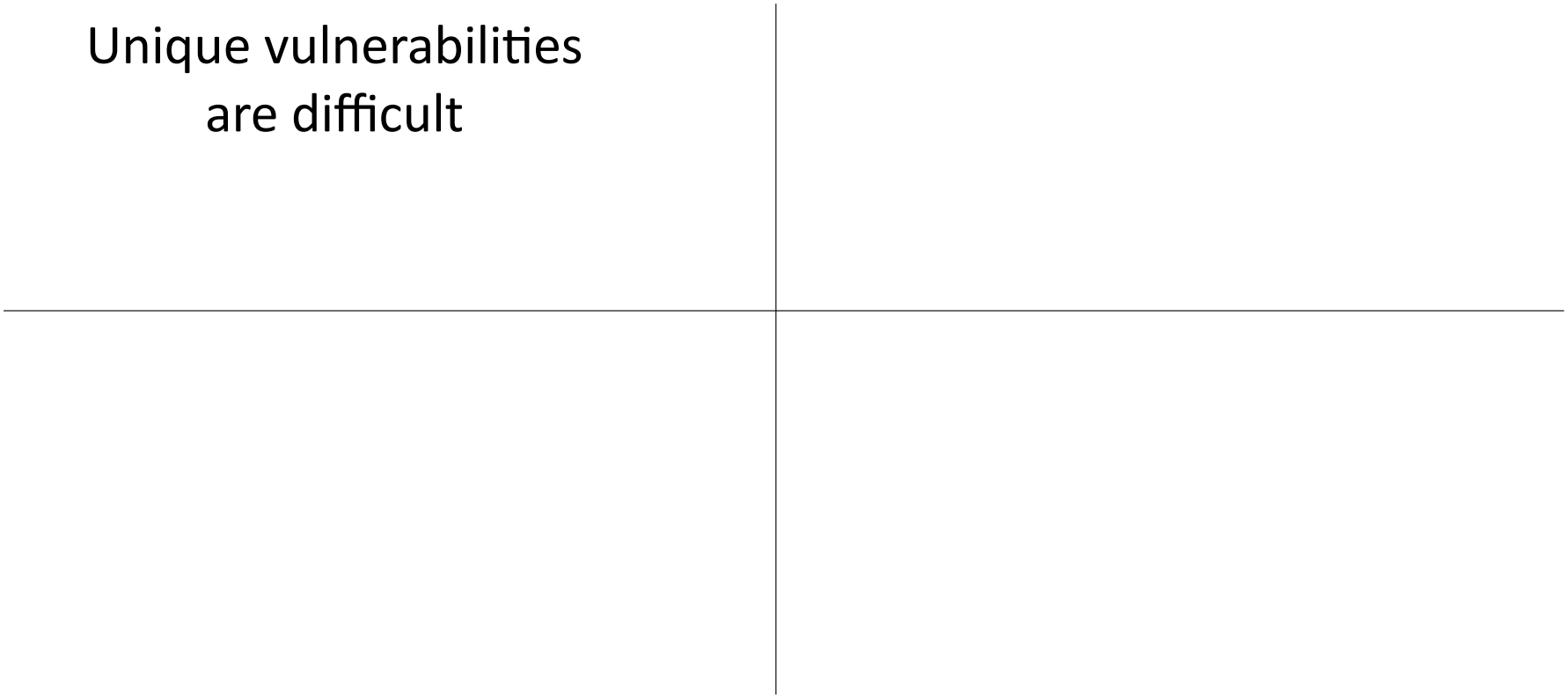
What do you do with vulnerabilities?

Coordinated Disclosure

Your Response Mileage May Vary!

- As you can fix this in the Unicode spec, the language, or the editor, you learn who cares!
  - Rust – very eager to fix the language spec
  - Java – Oracle said it was the editor's problem
- First responders often dismissed the problem (bug bounty contractors especially)
- In practice it took publicity + contacts + patience

Unique vulnerabilities are difficult



Unique vulnerabilities are difficult

CVEs are helpful

Unique vulnerabilities are difficult

CVEs are helpful

Maybe the government can help

Unique vulnerabilities are difficult

CVEs are helpful

Maybe the government can help

Linux is your friend

KrebsonSecurity
In-depth security news and investigation

HOME ABOUT THE AUTHOR ADVERTISING/SPEAKING

'Trojan Source' Bug Threatens the Security of All Code

November 1, 2021 54 Comments

Virtually all compilers — programs that transform human-readable source code into computer-executable machine code — are vulnerable to an insidious attack in which an adversary can introduce targeted vulnerabilities into any software without being detected, new research released today warns. The

Mailing List

Subscribe here

Schneier on Security

Blog Newsletter Books Essays News Talks Academic About Me

Home > Blog

Hiding Vulnerabilities in Source Code

Really interesting research demonstrating how to hide vulnerabilities in source code by manipulating how Unicode text is displayed. It's really clever, and not the sort of attack one would normally think

ZDNet

Trending Innovation Security Business Finance Education Home & Office More

MUST READ: Remote work or back to the office? The calculation just shifted again

Programming languages: This sneaky trick could allow attackers to hide 'invisible' vulnerabilities in code

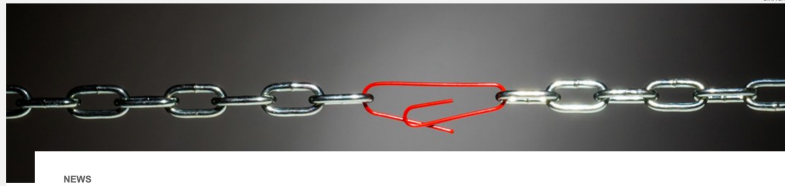
Rust programming language project has an update that addresses and more.

BLEEPINGCOMPUTER

NEWS DOWNLOADS

ComputerWeekly.com

IT Management Industry Sectors Technology Topics Search Compu



NEWS

Businesses and governments urged to take action over Trojan Source supply chain attacks

Businesses and governments have been put on alert to guard against Trojan Source hacking attacks

PEP 672 – Unicode-related Security Considerations for Python

Security advisory for rustc (CVE-2021-42574)

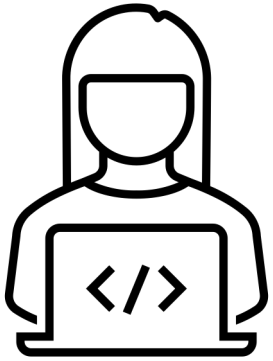
-source code

GIZMODO Tech. Science. Culture.

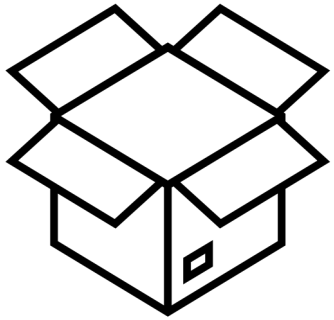
0

HOME LATEST TECH REVIEWS HOW TO SCIENCE SPACEFLIGHT

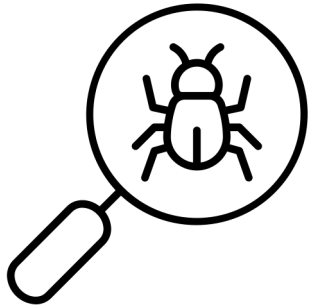
Source Code Defenses



Code Editors
& Repositories

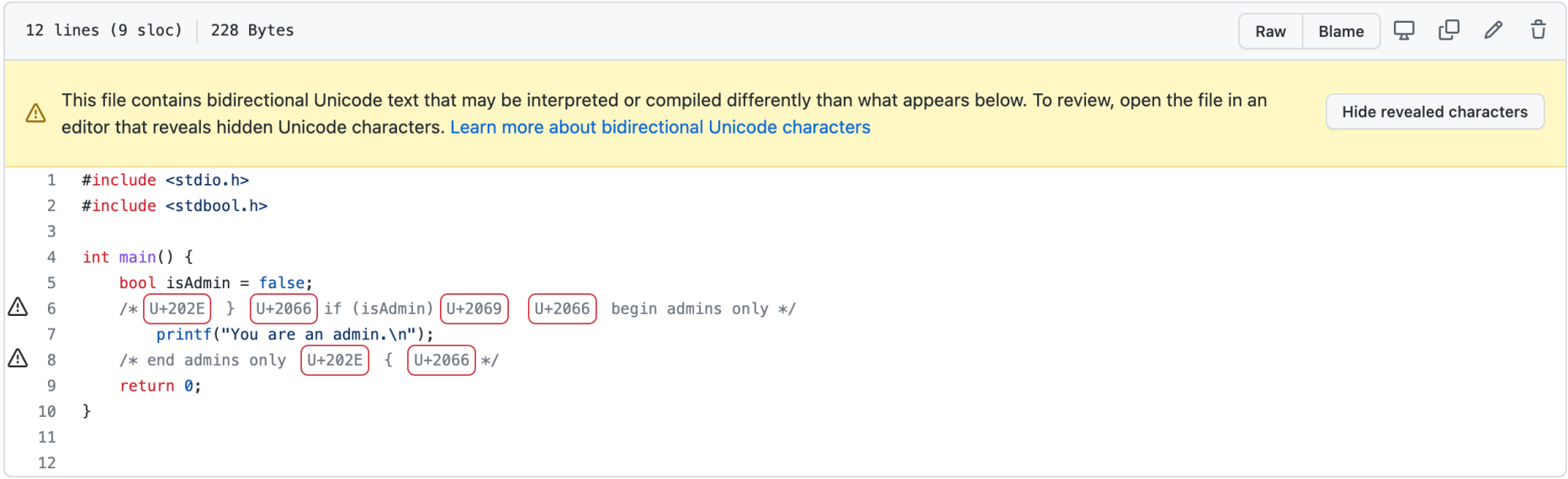


Compilers

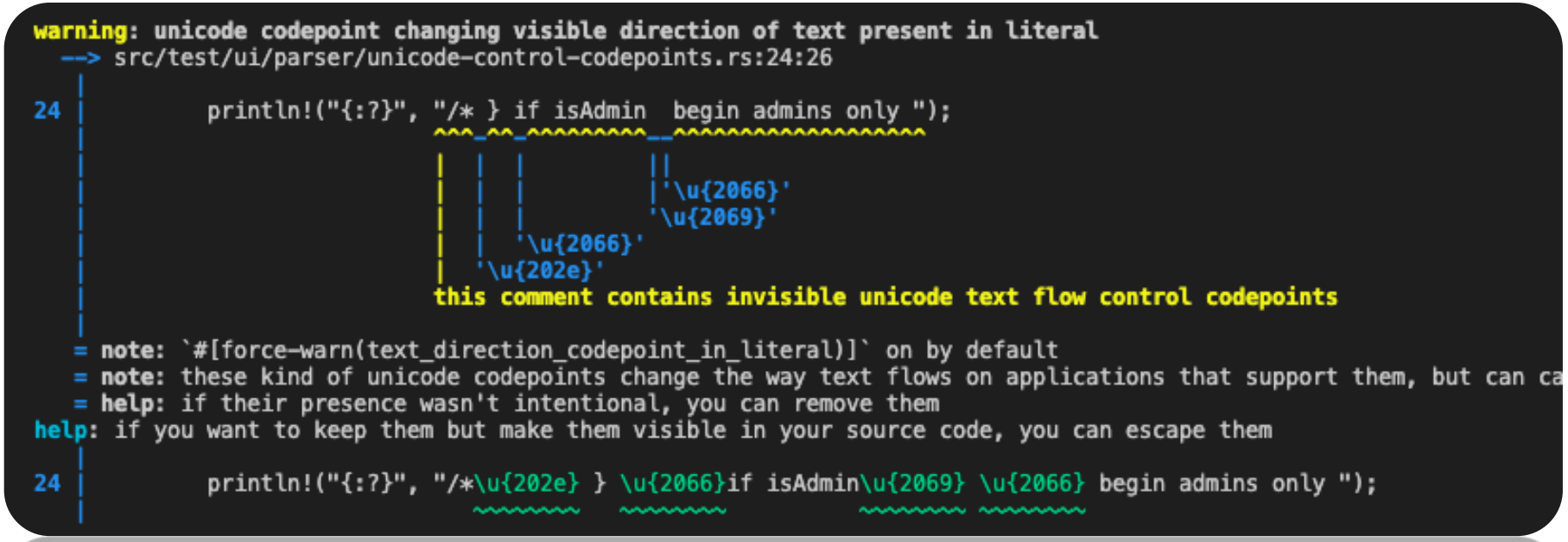


Static Code Analysis

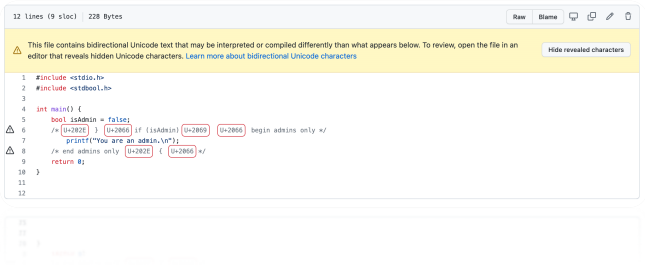
GitHub


12 lines (9 sloc) | 228 Bytes
Raw
Blame
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
Hide revealed 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;
}

Rust

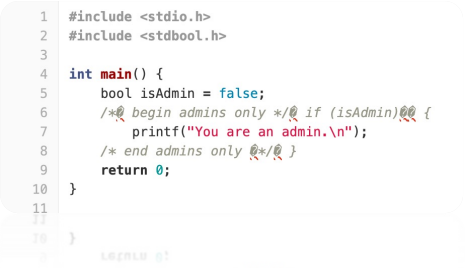

warning: unicode codepoint changing visible direction of text present in literal
--> src/test/ui/parser/unicode-control-codepoints.rs:24:26
println!("{:?}", "/* } if isAdmin begin admins only ");
'\u{2066}'
'\u{2069}'
'\u{2066}'
'\u{202e}'
this comment contains invisible unicode text flow control codepoints
= 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
println!("{:?}", "/*\u{202e} } \u{2066}if isAdmin\u{2069} \u{2066} begin admins only ");

GitHub

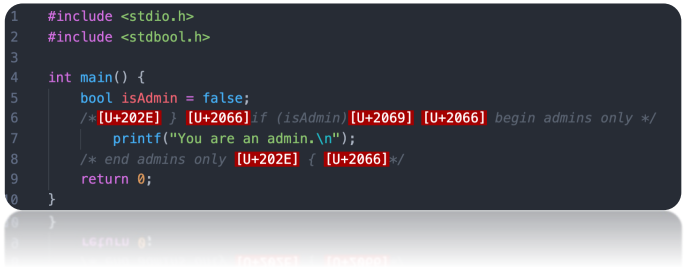


Bitbucket

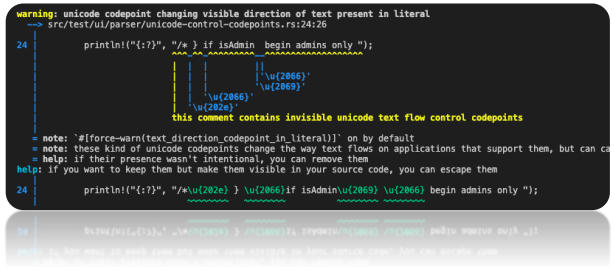
GitLab



VS Code



Rust



GCC

ML Defenses

Sanitize Inputs

ML versus Trojan Source

- Most languages / editors released fixes for the Trojan Source within months of disclosure
- But so far only one big NLP service has done anything to mitigate the attack on ML models (hat tip, Google!)
- Yet we discovered – and disclosed – the Bad Characters attack before the Trojan Source attack
- So what might be going on here?

So what's going on?

- Maybe ML models are too expensive to update? But you can sanitise the inputs easily enough
- Do ML vendors not know they need to do this? Surely not if they're IBM, MS, Google...
- At one, ML / security teams blamed each other
- Security, and safety, are whole-system properties!
- Other ML teams also tend to ignore this...

Topics for future research on code vs ML

- Cost of an upgrade / bugfix
- Time to do an upgrade / bugfix
- Culture of C coders versus data scientists
- Expectations of dependability
- Publicity for code bugs versus ML misbehaviour
- Competition / market power
- Maturity of technology and market

Takeaways

- Unicode makes it easy to hide invisible vulnerabilities in source code
- ... and to break text-based machine learning
- Defenses exist for both
- Why is the machine-learning community not listening?

More Info

Trojan Source Attack

https://trojansource.codes

Machine Learning Attack

https://imperceptible.ml