'Trojan Source' Bug Threatens the Security of All Code

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 vulnerability disclosure was coordinated with multiple organizations, some of whom are now releasing updates to address the security weakness.

Researchers with the **University of Cambridge** discovered a bug that affects most computer code compilers and many software development environments. At issue is a component of the digital text encoding standard [Unicode](), which allows computers to exchange information regardless of the language used. Unicode currently defines more than

143,000 characters across 154 different language scripts (in addition to many non-script character sets, such as emojis).

Specifically, the weakness involves Unicode’s bi-directional or “Bidi” algorithm, which handles displaying text that includes mixed scripts with different display orders, such as Arabic — which is read right to left — and English (left to right).

But computer systems need to have a deterministic way of resolving conflicting directionality in text. Enter the “Bidi override,” which can be used to make left-to-right text read right-to-left, and vice versa.

“In some scenarios, the default ordering set by the Bidi Algorithm may not be sufficient,” the Cambridge researchers wrote. “For these cases, Bidi override control characters enable switching the display ordering of groups of characters.”

Bidi overrides enable even single-script characters to be displayed in an order different from their logical encoding. As the researchers point out, this fact has previously been exploited to disguise the file extensions of malware disseminated via email.

Here’s the problem: Most programming languages let you put these Bidi overrides in comments and strings. This is bad because most programming languages allow comments within which all text — including control characters — is ignored by compilers and interpreters. Also, it’s bad because most programming languages allow string literals that may contain arbitrary characters, including control characters.

“So you can use them in source code that appears innocuous to a human reviewer [that] can actually do something nasty,” said **Ross Anderson**, a

This vulnerability is, as far as I know, the first one to affect almost *everything."*

professor of computer security at Cambridge and co-author of the research. "That's bad news for projects like Linux and Webkit that accept contributions from random people, subject them to manual review, then incorporate them into critical code. This vulnerability is, as far as I know, the first one to affect almost everything."

The research paper, which dubbed the vulnerability "Trojan Source," notes that while both comments and strings will have syntax-specific semantics indicating their start and end, *these bounds are not respected by Bidi overrides*. From the paper:

> "Therefore, by placing Bidi override characters exclusively within comments and strings, we can smuggle them into source code in a manner that most compilers will accept. Our key insight is that we can reorder source code characters in such a way that the resulting display order also represents syntactically valid source code."
>
> "Bringing all this together, we arrive at a novel supply-chain attack on source code. By injecting Unicode Bidi override characters into comments and strings, an adversary can produce syntactically-valid source code in most modern languages for which the display order of characters presents logic that diverges from the real logic. In effect, we anagram program A into program B."

Anderson said such an attack could be challenging for a human code reviewer to detect, as the rendered source code looks perfectly acceptable.

"If the change in logic is subtle enough to go undetected in subsequent testing, an adversary could introduce targeted vulnerabilities without being detected," he said.

Equally concerning is that Bidi override characters *persist through the* *copy-and-paste functions on most modern browsers, editors, and operating systems.*

"Any developer who copies code from an untrusted source into a protected code base may inadvertently introduce an invisible vulnerability," Anderson told KrebsOnSecurity. "Such code copying is a significant source of real-world security exploits."

Matthew Green, an associate professor at the **Johns Hopkins Information Security Institute**, said the Cambridge research clearly shows that most compilers can be tricked with Unicode into processing code in a different way than a reader would expect it to be processed.

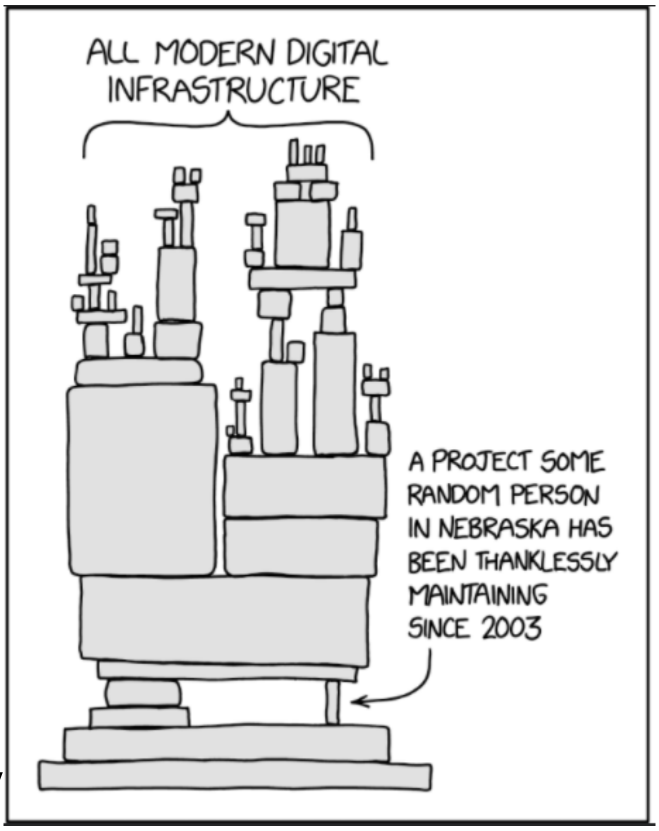
ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Image: XKCD.com/2347/

"Before reading this paper, the idea that Unicode could be exploited in some way wouldn't have surprised me," Green told KrebsOnSecurity. "What does surprise me is how many compilers will happily parse Unicode without any defenses, and how effective their right-to-left encoding technique is at sneaking code into codebases. That's a really clever trick I didn't even know was possible. Yikes."

Green said the good news is that the researchers conducted a widespread vulnerability scan, but were unable to find evidence that anyone was exploiting this. Yet.

"The bad news is that there were no defenses to it, and now that people know about it they might start exploiting it," Green said. "Hopefully compiler and code editor developers will patch this quickly! But since some people don't update their development tools regularly there will be some risk for a while at least."

Nicholas Weaver, a lecturer at the computer science department at **University of California, Berkeley**, said the Cambridge research presents "a very simple, elegant set of attacks that could make supply chain attacks much, much worse."

"It is already hard for humans to tell 'this is OK' from 'this is evil' in source code," Weaver said. "With this attack, you can use the shift in directionality to change how things render with comments and strings so that, for example 'This is okay" is how it renders, but 'This is' okay is how it exists in the code. This fortunately has a very easy signature to scan for, so compilers can [detect] it if they encounter it in the future."

The latter half of the Cambridge paper is a fascinating case study on the complexities of orchestrating vulnerability disclosure with so many affected programming languages and software firms. The researchers said they offered a 99-day embargo period following their initial disclosure to allow affected products to be repaired with software updates.

"We met a variety of responses ranging from patching commitments and bug bounties to quick dismissal and references to legal policies," the researchers wrote. "Of the nineteen software suppliers with whom we engaged, seven used an outsourced platform for receiving vulnerability disclosures, six had dedicated web portals for vulnerability disclosures, four accepted disclosures via PGP-encrypted email, and two accepted disclosures only via non-PGP email. They all confirmed receipt of our disclosure, and ultimately nine of them committed to releasing a patch."

Eleven of the recipients had bug bounty programs offering payment for vulnerability disclosures. But of these, only five paid bounties, with an average payment of $2,246 and a range of $4,475, the researchers reported.

Anderson said so far about half of the organizations maintaining the affected computer programming languages contacted have promised patches. Others are dragging their feet.

“We’ll monitor their deployment over the next few days,” Anderson said. “We also expect action from Github, Gitlab and Atlassian, so their tools should detect attacks on code in languages that still lack bidi character filtering.”

As for what needs to be done about Trojan Source, the researchers urge governments and firms that rely on critical software to identify their suppliers’ posture, exert pressure on them to implement adequate defenses, and ensure that any gaps are covered by controls elsewhere in their toolchain.

"The fact that the Trojan Source vulnerability affects almost all computer languages makes it a rare opportunity for a system-wide and ecologically valid cross-platform and cross-vendor comparison of responses," the paper concludes. "As powerful supply-chain attacks can be launched easily using these techniques, it is essential for organizations that participate in a software supply chain to implement defenses."

Weaver called the research "really good work at stopping something before it becomes a problem."

"The coordinated disclosure lessons are an excellent study in what it takes to fix these problems," he said. "The vulnerability is real but also highlights the even larger vulnerability of the shifting stand of dependencies and packages that our modern code relies on."

Rust has released a security advisory for this security weakness, which is being tracked as CVE-2021-42574 and CVE-2021-42694. Additional security advisories from other affected languages will be added as updates here.

The Trojan Source research paper is available here (PDF).