# TTP Detail – T1564.014

## TTP Information

Name: Extended Attributes

Description: Adversaries may abuse extended attributes (xattrs) on macOS and Linux to hide their malicious data in order to evade detection. Extended attributes are key-value pairs of file and directory metadata used by both macOS and Linux. They are not visible through standard tools like `Finder`, `ls`, or `cat` and require utilities such as `xattr` (macOS) or `getfattr` (Linux) for inspection. Operating systems and applications use xattrs for tagging, integrity checks, and access control. On Linux, xattrs are organized into namespaces such as `user.` (user permissions), `trusted.` (root permissions), `security.`, and `system.`, each with specific permissions. On macOS, xattrs are flat strings without namespace prefixes, commonly prefixed with `com.apple.\*` (e.g., `com.apple.quarantine`, `com.apple.metadata:\_kMDItemUserTags`) and used by system features like Gatekeeper and Spotlight.(Citation: Establishing persistence using extended attributes on Linux)

An adversary may leverage xattrs by embedding a second-stage payload into the extended attribute of a legitimate file. On macOS, a payload can be embedded into a custom attribute using the `xattr` command. A separate loader can retrieve the attribute with `xattr -p`, decode the content, and execute it using a scripting interpreter. On Linux, an adversary may use `setfattr` to write a payload into the `user.` namespace of a legitimate file. A loader script can later extract the payload with `getfattr --only-values`, decode it, and execute it using bash or another interpreter. In both cases, because the primary file content remains unchanged, security tools and integrity checks that do not inspect extended attributes will observe the original file hash, allowing the malicious payload to evade detection.(Citation: Low GroupIB xattrs nov 2024)

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Kill Chain Phases

**•** mitre-attack: defense-evasion