

# Detection Strategy for Compile After Delivery - Source Code to Executable Transformation, Detection Strategy DET0501

Archived: 2026-04-05 14:43:22 UTC

## AN1381

Detects compilation activity using `csc.exe`, `ilasm.exe`, or `msbuild.exe` initiated by user-space processes outside typical development environments, followed by execution or network activity from newly written binaries.

### Log Sources

### Mutable Elements

Field	Description
ParentProcessName	Filter for unexpected users (non-dev) launching compilers like <code>csc.exe</code> or <code>msbuild.exe</code>
OutputDirectoryPath	Adjust paths for sensitive file write zones (e.g., <code>C:\Users\Public\`</code> , <code>%TEMP%`</code> , or Desktop)
TimeWindow	Tune the correlation window between compilation and subsequent execution or C2

## AN1382

Detects GCC or Clang invoked on suspicious file paths (e.g., `/tmp/`, `~/Downloads`) with output to executable binaries, followed by execution or outbound traffic from these binaries.

### Log Sources

### Mutable Elements

Field	Description
CompilerBinaryPath	Specify path and binaries for detection (e.g., <code>/usr/bin/gcc`</code> , <code>/opt/mingw/bin/gcc`</code> )
FilePermissionProfile	Match uncommon <code>chmod</code> behavior post-compilation (e.g., <code>+x</code> in <code>/tmp`</code> or home directories)

## AN1383

Detects non-standard compilation activity via Xcode CLI tools or bundled GCC/MONO packages writing new executable files and executing them outside dev environments (e.g., user Downloads folder).

## Log Sources

## Mutable Elements

Field	Description
CompilerInvocationPattern	Detect calls to `xcodebuild`, `clang`, or `/Applications/Mono.app/...` from non-admin users
OutputBinaryPath	Monitor for output files in user-writable paths (e.g., `~/Library/Caches`, `~/Downloads`)

---

Source: <https://attack.mitre.org/detectionstrategies/DET0501>