Detect Adversary Deobfuscation or Decoding of Files and Payloads, Detection Strategy DET0275

Archived: 2026-04-02 11:32:27 UTC

AN0767

An adversary leverages built-in tools such as certutil.exe, powershell.exe, or copy.exe to decode, reassemble, or extract hidden malicious content from obfuscated containers or encoded formats. The decoding utility often spawns shortly after file staging or download and may be chained with script interpreters or further payload execution.

Log Sources

Mutable Elements

Field	Description
ToolName	May vary across environments (e.g., certutil, powershell, copy, expand, 7zip)
FileExtensionFilter	Targets may use .txt, .cer, .enc, .b64, .zip, etc. to disguise payloads
CommandLineRegex	Command syntax varies between base64 decoding, copy /b, and expand switches
TimeWindow	Deobfuscation typically follows staging/download within a short timeframe

AN0768

The adversary uses native utilities like base64, gzip, tar, or openssl to decode, decompress, or decrypt files that were previously staged or downloaded. These tools may be chained with curl/wget and executed via bash/zsh, often to extract an embedded payload or reverse shell script.

Log Sources

Data Component	Name	Channel
Command Execution (DC0064)	auditd:SYSCALL	bash/zsh of base64, tar, gzip, or openssl immediately after file write

Mutable Elements

Field	Description
ShellProcessName	Shell interpreter may vary (bash, zsh, dash, sh)
DecodeUtility	May include base64, openssl, gunzip, tar, uudecode
ParentProcess	Expected parent process may vary in attacker chain (e.g., curl, bash, ssh)
ArgumentPattern	Detection regex should support flexible patterning of decode switches

AN0769

The adversary invokes built-in scripting or decoding tools like base64, plutil, or AppleScript-based utilities to decode files embedded in staging artifacts. Decoding often occurs post-download or as part of post-exploitation payload deployment via zsh, python, or osascript.

Log Sources

Mutable Elements

Field	Description
DecodeInterpreter	Could involve base64, osascript, python, perl, or plutil
ExecutionContext	Deobfuscation may happen within GUI app context or LaunchAgent/Daemon
UserContext	May differ depending on local user, root escalation, or persistence method

---