HashEye

// SMART CONTRACT SECURITY AUDIT

Lisk SDK 6.1 modules

Security assessment by HashEye · prepared for Lisk

HASHEYE AUDITED

PROJECT	Lisk SDK 6.1 modules
CLIENT	Lisk
CATEGORY	Blockchain
PUBLISHED	September 1, 2023
REPORT ID	research-lisk-sdk-6-1-modules-2023-09-01-1o4hof

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-lisk-sdk-6-1-modules-2023-09-01-1o4hof.

LiskSDK6.1Sapphire,NFTandPoA modules SecurityAssessment November27,2023 Preparedfor: OliverBeddows LiskFoundation Preparedby:VascoFranco,SamAlws,andPaweł Płatek

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 1LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

NoticesandRemarks CopyrightandDistribution ©2023byhasheye,Inc. Allrightsreserved.hasheyeherebyassertsitsrighttobeidentifiedasthecreatorofthis reportintheUnitedKingdom. Thisreportisconsideredbyhasheyetobepublicinformation;itislicensedtoLisk Foundationunderthetermsoftheprojectstatementofworkandhasbeenmadepublicat LiskFoundation’srequest.Materialwithinthisreportmaynotbereproducedordistributed inpartorinwholewithouttheexpresswrittenpermissionofhasheye. Thesolecanonicalsourceforhasheyepublicationsisthehasheye Publicationspage. 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 2LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

B.CodeMaturityCategories49 C.CodeQualityRecommendations51 NFTModule51 hasheye 3LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

PoAModule52 D.StaticAnalysisToolConfiguration55 E.FixReviewResults57 DetailedFixReviewResults58 F.FixReviewStatusCategories60 hasheye 4LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

ProjectSummary ContactInformation Thefollowingmanagerswereassociatedwiththisproject: DanGuido,AccountManagerSamGreenup,ProjectManager dan@hasheye.iosam.greenup@hasheye.io Thefollowingengineerswereassociatedwiththisproject: VascoFranco,ConsultantSamAlws,Consultant vasco.franco@hasheye.iosam.alws@hasheye.io PawełPłatek,Consultant pawel.platek@hasheye.io ProjectTimeline Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent August25,2023Pre-projectkickoffcall September1,2023Statusupdatemeeting#1 September11,2023Deliveryofcomprehensivereportdraft September11,2023Reportreadoutmeeting November27,2023Deliveryofcomprehensivereport hasheye 5LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

ExecutiveSummary EngagementOverview LiskFoundationengagedhasheyetoreviewthesecurityoftheLiskSDKv6.1NFTand PoAmodules.TheNFTmoduleimplementstheLiskprotocol’slogicforhandlingNFTs, includingintra-andcross-chaintransfers.ThePoAmodulemanagesandupdatesthe validatorlistforproof-of-authoritychains. Ateamofthreeconsultantsconductedthereviewfromaugust28toSeptember11,2023, foratotaloffiveengineer-weeksofeffort.Ourtestingeffortsfocusedontheneftandpoa modulesoftheLiskSDKv6.1.Withfullaccesstosourcecodeanddocumentation,we performedstaticanddynamictestingoftheseLiskSDKmodules,usingautomatedand manualprocesses. ObservationsandImpact Duringtheaudit,wefoundtwoissuesofmediumseverity: • TOB-LISK2-14:UsersmaybypassanNFTscreationfeebyfakingbouncedmessages. • TOB-LISK2-10:Thefunctionthatremovesallsupportednftsmayfailtoremove supportforallNFTs. Wealsofoundanumberoflow-severityandinformationalfindings,mostrelatedtodata validation. Recommendations Basedonthecodebasematurityevaluationandfindingsidentifiedduringthesecurity review,hasheyerecommendsthatLisktakethefollowingsteps: • Remediatethefindingsdisclosedinthisreport.Thesefindingsshouldbe addressedaspartofadirectremediationoraspartofanyrefactorthatmayoccur whenaddressingotherrecommendations. • Improvetesting.Whileunittestingcoverageisgood,werecommendaddingfuzz testingtosomefunctionality,asisrecommendedinfindingTOB-LISK2-15,and includingthestaticanalysisrulesweprovidedinthefirstphaseoftheaudit,which wouldhelpfindissuessuchasTOB-LISK2-8.Finally,edgecases(e.g.,minimumand maximumintegervaluesinschemaencodinganddecoding)shouldbemore thoroughlytested. hasheye 6LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Thefollowingtablesprovidethenumberoffindingsbyseverityandcategory. EXPOSUREANALYSIS SeverityCount High0 Medium2 Low6 Informational9 Undetermined0 CATEGORYBREAKDOWN CategoryCount AuditingandLogging2 Configuration3 DataValidation11 Testing1 hasheye 7LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

ProjectGoals Theengagementwasscopedtoprovideasecurityassessmentoftheliskdknftandpoa modules.Specifically,wesoughttoanswerthefollowingnon-exhaustivelistofquestions: • DoestheNFTmoduleimplementationmatchitsLIPspecification? • Canattackersarbitrarilymintordestroynfts? • CanattackerstransferNFTsintra-orcross-chainwithoutowningtheNFT? • CanattackersbypasspayingtheNFTcreationfee? • DoesthePoAmoduleimplementationmatchitsLIPspecification? • CanattackerstakecontrolofaPoAchain? • CanPoAsignaturesbemodifiedorreusedtoperformunexpectedauthority updates? • ArePoAvalidatorlistupdatesperformedcorrectlyandsafely? hasheye 8LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthetargetslistedbelow. LiskSDK Repositoryhttps://github.com/LiskHQ/lisk-sdk/releases/tag/v6.1.0-beta.0 Version ed5649eb954c7c47e11eb2d2ea2b84b9336c4c4b TypeTypeScript PlatformAny LiskImprovementProposals(LIPs) Repositoryhttps://github.com/LiskHQ/lips Version c7d59b177cdd74553d5995298afce3a835dd67f7 TypeDocumentation PlatformN/A hasheye 9LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

ProjectCoverage Thissectionprovidesanoverviewoftheanalysiscoverageofthereview,asdeterminedby ourhigh-levelengagementgoals.Ourapproachesincludedthefollowing: • ManualreviewoftheNFTandPoAmodules,comparingtheimplementationagainst theLIPspecifications • UseofstaticanalysistoolssuchasSemgrepandCodeQLontheNFTandPoA modules(includingtheSemgreprulesprovidedinthelastauditoftheLiskSDK) • Dynamicreviewofsomeoftheneftandpoamodule’sfunctionalitywiththeuseof theprovidedLiskSDKexamples • Abriefmanualreviewofthecodepathsin lisk-codec , lisk-cryptography ,and lisk-validator relevanttoPoAandNFTfunctionality hasheye 10LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

AutomatedTesting hasheyeusesautomatedtechniquestoextensivelytestthesecuritypropertiesof software.Weusebothopen-sourcestaticanalysisandfuzzingutilities,alongwithtools developedinhouse,toperformautomatedtestingofsourcecodeandcompiledsoftware. TestHarnessConfiguration Weusedthefollowingtoolsintheautomatedtestingphaseofthisproject: ToolDescriptionPolicy SemgrepAnopen-sourcestaticanalysistoolforfindingbugsand enforcingcodestandardswheneditingorcommittingcode andduringbuildtime AppendixD CodeQLAcodeanalysisenginedevelopedbyGitHubtoautomate securitychecks AppendixD TestResults Theresultsofthisfocusedtestingaredetailedbelow. PropertyToolResult Aneventclassdoesnothaveafunctioncaller error thatadds nonpersistentevents(error_event_is_not_revert rule). SemgrepTOB-LISK2-9 Astoredvariableisnotreadintoavariable,modified,andthen notstoredagain(get_modify_no_set_on_stores rule). SemgrepPassed Astoreoreventisnotregisteredwiththeincorrectclass (module_registration_of_correct_class rule). SemgrepPassed hasheye

Therearenottwostoresregisteredwiththesameindex (module_stores_same_index rule). SemgrepPassed Aschemaobjectdoesnotcontain minItems and/or maxItems onapropertythatisnotoftype array (schema_min_max_items_without_array rule). SemgrepPassed Allschemapropertieshaveafieldnumber (schema_property_element_without_field_number rule). SemgrepPassed Therearenottwopropertiesofthesameschemawiththe samefieldnumber (schema_with_duplicate_field_number rule). SemgrepPassed Schemasrequireeveryproperty (schema_with_field_not_required rule). SemgrepPassed Aschemadoesnotrequireapropertythatdoesnotexist (schema_with_required_that_is_not_a_property rule). SemgrepPassed Schemasusetheformatattributeonlyonnon-integertypes (schema_int_format_with_integer_type rule). SemgrepPassed hasheye

CodebaseMaturityEvaluation hasheyeusesatraffic-lightprotocoltoprovideeachclientwithaclearunderstandingof theareasinwhichitscodebaseismature,immature,orunderdeveloped.Deficiencies identifiedhereoftenstemfromrootcauseswithinthesoftwaredevelopmentlifecyclethat shouldbeaddressedthroughstandardizationmeasures(e.g.,theuseofcommonlibraries, functions,orframeworks)ortrainingandawarenessprograms. CategorySummaryResult ArithmeticWhennecessary,boundsarecheckedbycomparing numbersto INT_MAX and INT_MIN values.64-bitinteger arithmeticavoidsoverflowsbyusingthe BigInt type. Wefoundthattheschemavalidatorusesanincorrectvalueforthe MIN_SINT_32 constantinTOB-LISK2-7. Satisfactory AuditingLogqualitywouldbeimprovedifthePoA RegisterAuthority and UpdateGeneratorKey commandsweretoemiteventsduringexecution. WealsofoundthatintheNFTmodule,someerrorevents werenotpersisted(TOB-LISK2-8),andthatbouncedNFT transfereventscontainedincorrectandinsufficientdata (TOB-LISK2-12). Weak Authentication/ AccessControls Authenticationforcommandsistypicallydoneviachecks onthetransaction’ssender;wefoundnoissuesrelated tothis. Satisfactory Complexity Management Overall,PoAandNFTcodeiswell-organizedaccordingto astandardstructure. IntheNFTmodule,wefoundthattherewasduplicated codeofcriticalfunctionality(TOB-LISK2-2and TOB-LISK2-13),andextensiveuseofmagicvalues (TOB-LISK2-4),whichmadethecodemorecomplexthan necessary. Moderate Cryptography andKey Management NocryptographyisdoneintheNFTmodule.When cryptographyisdoneinthePoAmodule,librariesfrom otherpartsofthe lisk-sdk repositoryareused,which areout-of-scopeforthisaudit. Satisfactory hasheye

DataHandlingWefoundmultipleissuesrelatedtodatavalidation: whereitistoostrict(TOB-LISK2-1),whereitshouldbe stricter(TOB-LISK2-3,TOB-LISK2-6,TOB-LISK2-10, TOB-LISK2-13,TOB-LISK2-14,TOB-LISK2-15, TOB-LISK2-16, TOB-LISK2-17),andwhereitcouldbedone inamorerobustway(TOB-LISK2-5).Mostoftheseare low-severityorinformationalissues,butsomehavea largerpracticaloutcome:userswillsometimesbeunable tosendNFTscross-chain(TOB-LISK2-1),usersmay bypassNFTfeepayments(TOB-LISK2-14),andthe removeSupportAllNFTs maynotremovesupportforall NFTs(TOB-LISK2-10). Wedidnotfindanyissueswhereinsufficientdata validationallowsforalossofNFTsoralossofcontrol overaPoAchain. Moderate DocumentationBothmoduleshaveLIPsthatdescribetheiroperationin detail.TheseLIPsaregenerallywell-written,andthe modulecodegenerallymatchesupwiththe correspondingpseudocodeintheLIPs;however,there aresomesmallexceptions(seeappendixC). Satisfactory MaintenanceTheonlydirectdependenciesoftheNFTandPoA modulesareonotherpartsofthe lisk-sdk codebase, whichwascoveredinourpreviousauditwithLiskand wasthereforenotcoveredinthisaudit.ThePoAmodule indirectly(through @liskhq/lisk-cryptography)uses the @chainsafe/blst NodeJSlibraryforsignature verification;wefoundthatthecurrentversion(0.2.9)of thislibraryisused,butdidnotverifythesecurityofthis library. Satisfactory MemorySafety andError Handling TheLiskSDKisimplementedinamemory-safelanguage, somemorysafetyissueswerenotconsideredduringthe audit. Errorsareappropriatelyhandled,typicallybythrowing exceptionsorbyexitingearlyandlogginganerrorevent.

Wefoundasingleissuewhereamalformed NFTthatis beingrecoveredmaycauseanunexpectedcrash (TOB-LISK2-15). Satisfactory TestingandTheNFTandPoAmodulesarethoroughlytested.TestSatisfactory hasheye 14LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Verificationcoverageiscollectedoneachcommitusing Codecov; currently,onthedevelopmentbranch,codecovshows themoduleshaving97%and94%coverage,respectively. TheuseofTypeScriptverifiesthatJSdynamictypeerrors areavoided. However,ourbriefexaminationofthetestsrevealeda bugthatmadeoneofthetestsineffective (TOB-LISK2-11).Wealsofoundthatedgecaseswerenot alwaysthoroughlytested(TOB-LISK2-1,TOB-LISK2-10, TOB-LISK2-17,andmore). Furthermore,wethinktheSemgreprulesandfuzzers thatweprovidedduringthephaseoneauditshouldbe includedintheproject’sCI/CDpipeline.Usingtheserules, wefoundonebug(TOB-LISK2-8)thatcouldhavebeen prevented. Finally,wethinkfuzzingshouldbeimplementedforsome functionalitywhereitwouldhelptofindedgecases(e.g., TOB-LISK2-15andTOB-LISK2-17). hasheye 15LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails. IDTitleTypeSeverity 1Impossibletosendcross-chainNFTtransfersin certainconfigurations DataValidationLow 2NFTmodulehasrepeatedtransferfunctionalityConfigurationLow 3NFTlockmethoddoesnotcheckifmoduleis NFT_NOT_LOCKED DataValidationInformational 4NFTmethodsAPIcouldbeimprovedConfigurationInformational 5Checksdonemanuallyinsteadofusingschema validation DataValidationInformational 6Updateauthoritycommandallowsvalidatorsto have0weight DataValidationInformational 7IncorrectMIN_SINT_32constantDataValidationInformational 8EventerrorsarerevertedAuditingand Logging Low 9UnspecifiedandpoorlynamedNFTendpointsConfigurationInformational 10TheremoveSupportAllNFTsfunctionmaynot removesupportforallNFTs DataValidationMedium 11BuginremoveSupportAllNFTstestsTestingInformational 12BouncedNFTtransfersmaycauseeventsto containincorrectdata Auditingand Logging Low hasheye 16LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

13AnNFT'sattributesArraycanhaveduplicate modules DataValidationLow 14Bouncedmessagesmaybeabusedtobypass NFT fees DataValidationMedium 15NFTrecoverfunctionmaycrashDataValidationInformational 16NFTrecoverfunctiondoesproperlyvalidateNFT attributesarray DataValidationInformational 17Thecodec.encodefunctionencodeslargerthan expectedintegers DataValidationLow hasheye 17LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

DetailedFindings 1.Impossibletosendcross-chainNFTtransfersincertainconfigurations Severity:LowDifficulty:Low Type:DataValidationFindingID:TOB-LISK2-1 Target: lisk-sdk/framework/src/modules/nft/method.ts Description The isNFTSupported function—afunctioncalledintheCrossChainTransferCommand’s executefunctionwhenreceivingNFTsfromforeignchainstocheckifanNFTis supported—mayreturnincorrectresultsbecausethe getCollectionID function(figure 1.1)enforcesthatanNFTmustbestoredbeforereturningthecorrespondingcollectionID.

```
publicasyncgetCollectionID( methodContext:ImmutableMethodContext, nftID:Buffer, ):Promise<Buffer>{
constnftStore=this.stores.get(NFTStore); constnftExists=awaitnftStore.has(methodContext,nftID);
if(!nftExists){ thrownewError('NFTsubstoreentrydoesnotexist'); }
returnnftID.slice(LENGTH_CHAIN_ID,LENGTH_CHAIN_ID+LENGTH_COLLECTION_ID); }
```

Figure1.1:The getCollectionID function (lisk-sdk/framework/src/modules/nft/method.ts#187-197) The isNFTSupported functioncalls getCollectionID if:a)theNFTIDisnotnativetothe chain,b)the SupportedNFTsStore storedoesnothavethe ALL_SUPPORTED_NFTS_KEY key,andc)the supportedCollectionIDArray arrayforthegivenchainisnotempty (indicatingthatallcollectionsaresupported).Underthesecondition, isNFTSupported willalwaysthrowanexceptionandpreventthecrosschaintransfer. ExploitScenario AdeveloperconfigureschainAtosupportNFTsofcollection1fromchainB.Auserfrom chainBattemptstotransfertochainAanNFTbelongingtocollection1ofchainB.The cross-chainmessage(CCM)failsandtheNFTcannotbetransferred. hasheye 18LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Recommendations Shortterm,removetheNFTexistencecheckfromthe getCollectionID function.Thiswill allowobtainingthecollectionIDofnon-storedNFTIDs,whichisnecessaryinthe isNFTSupported function. Longterm,createteststhatcoverthiscaseandallotherpossibleconfigurationsofthe SupportedNFTsStore store. hasheye 19LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

2.NFTmodulehasrepeatedtransferfunctionality Severity:LowDifficulty:High Type:ConfigurationFindingID:TOB-LISK2-2 Target:NFTmodule Description TheNFTmodule’stransfercommandfunctions, verify and execute ,togetherhavethe samechecksandfunctioncallsasthe transfer method.Theirdifferenceisinhowthey handleerrors.Havingduplicatedcode,especiallyforcriticalfunctionalitysuchas

transferringNFTs,isbadpractice,asthecodemaydivergeandleadtoexploitablebugs.
Thesameproblemexistsonthe TransferCrossChainCommand commandand transferCrossChain method.
ExploitScenario Adeveloperfixesacriticalvulnerabilityinoneofthefunctionswithcopiedfunctionality,
butforgetstofixthecounterpart.Thecoderemainsvulnerable.Anattackerfindsoutand stealsNFTsfromusers.
Recommendations Shortterm,factorizethecodeintoasinglefunctionthatverifiesifatransferisvalid.The
commonfunctionshouldthrowcustomexceptionsbasedonthetypeoferrorthat
occurred.Inboththetransfercommandandmethod,havethecodecallthesinglefunction
andhandleerrorsappropriately(e.g.,byjustlettingtheexceptiongothrough,orby
catchingitandemittinganevent).
Longterm,reviewothermodulesforduplicatedfunctionality,especiallyincriticalcode
pathssuchasthecodethathandlestransfers.Ifthecodeisduplicated,factorizeintoa singlefunction. hasheye
20LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

3.NFTlockmethoddoesnotcheckifmoduleisNFT_NOT_LOCKED Severity:InformationalDifficulty:High
Type:DataValidationFindingID:TOB-LISK2-3 Target: lisk-sdk/framework/src/modules/nft/method.ts#L339-
L397 Description TheNFTlockmethoddoesnotcheckthatthemoduleattemptingtolocktheNFTisnot
equaltothemagicvaluethatrepresentsanunlockedNFT,the NFT_NOT_LOCKED constant (whichisequalto ‘nft’
).Itisunlikelythatanothermoduleisorcanbenamed ‘nft’ ;
however,thischeckshouldbeaddedtopreventmisuseofthe lock method. ExploitScenario
AnothermodulethatextendsormakesuseoftheNFTmodulecallsthe lock methodwith themoduleargumentsetto
‘nft’ bymistake(e.g.,adevelopermisunderstandsthe
purposeofthemoduleargument).TheNFTdoesnotgetlockedasthedeveloperexpected. Recommendations
Shortterm,addacheckinthecodethatensuresthatthe lock function’s module argumentisnotequalto
NFT_NOT_LOCKED . Longterm,reviewstoresthatuseasinglestatevariabletorepresentmorethanonestate
throughtheuseof“magic”values.Intheexampledescribedabove,the lockingModule
statevariableisusedtodeterminewhichmodulelockedtheNFTandhasaspecial value— NFT_NOT_LOCKED —
toindicatethatnomodulelockedit.Ifpossible,werecommend storingthestateintwovariables, isLocked and
lockingModule ,toreducetheuseof
“magicvalues”thatmayleadtostateconfusion.Dothesamedecompositionforother
stores,whereapplicable.Ifthesechangesarenotpossibleorwanted,ensurethatthe
magicvaluesarealwayscheckedwhenthestateismodified. hasheye
21LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

4.NFTmethodsAPIcouldbeimproved Severity:InformationalDifficulty:High
Type:ConfigurationFindingID:TOB-LISK2-4 Target:NFTmethods Description
TheNFTmoduleperformsvariouschecks,suchasifanNFTisescrowedorifanNFTis
lockedinawaythatmakesthecodelongerandlessreadable.Somechecks(e.g.,NFT
existence)alsohappenmorethannecessarybecauseofthewaytheAPIisdesigned.
TocheckifanNFTisescrowedtoanotherchain,thecodedoesthecheckshowninfigure 4.1inseverallocations.
if(owner.length===LENGTH_CHAIN_ID){ Figure4.1:An if statementthatchecksifanNFTisescrowed (lisk-
sdk/framework/src/modules/nft/commands/transfer.ts#60) Thischeckreliesonthefactthatthe owner
fieldwillhaveadifferentlengthifitisstoring
theaddressofanativeuserorthechainIDofaforeignchain.Tosomeonenotfamiliarwith
theinnerworkingsoftheNFTmodule(e.g.,anewemployeeoracurioususer),thisisnot
clear.Instead,thecodeshouldcallafunctionnamed isNFTEscrowed toclarifythe
purposeofthecheckbeingperformed.
TocheckifanNFTislocked,thecodedoesthecheckshowninfigure4.1inseveral locations.
constlockingModule=awaitthis._method.getLockingModule(context.getMethodContext(), params.nftID,);
if(lockingModule!==NFT_NOT_LOCKED){ Figure4.2:An if statementthatchecksifanNFTislocked (lisk-
sdk/framework/src/modules/nft/commands/transfer.ts#68-75) Thischeckreliesonthefactthatthe
lockingModule hasa“magicvalue,” NFT_NOT_LOCKED
,thatrepresentsanunlockedNFT.Again,tosomeonenotfamiliarwith
theinnerworkingsoftheNFTmodule,itmaynotbeclearwhythecodeischeckingthe hasheye
22LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

lockingModule toseeifanNFTislocked.Instead,thecodecouldcallafunctionnamed isNFTLocked
.Thiswouldalsoshortenthecode’slength.
AnotherproblemwiththecurrentAPIisthatallNFTmethodstakeanNFTIDargument;as
aresult,manyfunctionsendupperformingrepeatedoperationswhencalledinsuccession. Oneexampleofthisisthe
verify functionofthe TransferCommand command(figure 4.3).
publicasyncverify(context:CommandVerifyContext<Params>): Promise<VerificationResult>{
const{params}=context; constnftStore=this.stores.get(NFTStore);
constnftExists=awaitnftStore.has(context,params.nftID); if(!nftExists){
thrownewError('NFTsubstoreentrydoesnotexist'); }
constowner=awaitthis._method.getNFTOwner(context.getMethodContext(), params.nftID);
if(owner.length===LENGTH_CHAIN_ID){ thrownewError('NFTisescrowedtoanotherchain'); }

if(!owner.equals(context.transaction.senderAddress)){
thrownewError('TransfernotinitiatedbytheNFTowner'); }
constlockingModule=awaitthis._method.getLockingModule(context.getMethodContext(), params.nftID,);
if(lockingModule≠=NFT_NOT_LOCKED){ thrownewError('LockedNFTscannotbetransferred'); }
return{status:VerifyStatus.OK,}; } Figure4.3:Theverifyfunctionofthe TransferCommand
command(shortenedforbrevity) (lisk-sdk/framework/src/modules/nft/commands/transfer.ts#45-80)
Thecodefromfigure4.3callsthe nftStore.has functiontocheckiftheNFTexists.Then, the getNFTOwner
callinternallycalls nftStore.has againtocheckiftheNFTexistsandit calls nftStore.get
toobtainthedataoftheNFTandextractitsowner.Finally, getLockingModule willinternallycall getNFTOwner
(whichcalls nftStore.has and nftStore.get again).Inconclusion,inthesethreeoperations—
checkingtheexistenceof theNFT,gettingitsowner,andgettingitslockingmode—theexistencecheckisperformed
threeseparatetimes,andtheNFTdataisobtainedtwotimes. hasheye
23LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

IftheAPIwasdesignedtoreceivetheNFTdataobjectinsteadoftheNFTIDformost
operations(allexceptcheckinganNFT’sexistenceandgettingtheNFT),thenthis
duplicationofcheckswouldnotoccur. Infigure4.4,wesuggestanalternativeimplementationthatcallsthe
isNFTEscrowed and isNFTLocked forshorterandcleanercode.Thisimplementationpassestothesefunctions
theNFTobjectinsteadoftheNFTIDtoavoidtheduplicationofchecksdescribedabove.
publicasyncverify(context:CommandVerifyContext<Params>): Promise<VerificationResult>{
const{params}=context; constctx=context.getMethodContext();
constnftData=awaitthis._method.getNFT(ctx,params.nftID); if(!nftData){
thrownewError('NFTsubstoreentrydoesnotexist'); } if(this._method.isNFTEscrowed(ctx,nftData)){
thrownewError('NFTisescrowedtoanotherchain'); }
if(!nftData.owner.equals(context.transaction.senderAddress)){
thrownewError('TransfernotinitiatedbytheNFTowner'); } if(this._method.isNFTLocked(ctx,nftData)){
thrownewError('LockedNFTscannotbetransferred'); } return{status:VerifyStatus.OK,}; }
Figure4.4:Analternativeexampleimplementationoftheverifyfunctionofthe TransferCommand command
Recommendations Shortterm,createmethodstocheckifanNFTisescrowedandifanNFTislocked.Thiswill
improvethecode’sreadabilityandreduceitssize.Furthermore,considerupdatingmost
methodsoftheNFTmoduletoreceivetheNFTobjectinsteadoftheNFTID.Thiswillreduce
theamountoftimethesamechecksdoneineachfunction.
Longterm,reviewothermoduleswheretheserecommendationsmayalsoapply. hasheye
24LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

5.ChecksdonemanuallyinsteadofusingschemavalidationSeverity:InformationalDifficulty:High
Type:DataValidationFindingID:TOB-LISK2-5 Target: lisk-sdk/framework/src/modules/poa/schemas.ts
Description Intwolocationsintheproofofauthoritymodule,checksaredonemanuallywhenschema
validationcouldbeusedinstead.Usingschemavalidationwouldresultinmorerobust code. In endpoint.ts
,theparametergivento getValidator ischeckedinthefollowingway: const{address}=context.params;
if(typeofaddress≠='string'){thrownewError('Parameteraddressmustbeastring.'); }
cryptoAddress.validateLisk32Address(address); Figure5.1:Checkin getValidator (lisk-
sdk/framework/src/modules/poa/endpoint.ts#33-37) However,thefollowingcheckcouldbeperformedinstead:
validator.validate(getValidatorRequestSchema,address) Figure5.2:Proposedcheckfor getValidator In
commands/update_authority.ts ,thelengthofthecommand’s newValidators listis
checkedinthefollowingway: if(newValidators.length<1||newValidators.length>MAX_NUM_VALIDATORS){
thrownewError(`newValidatorslengthmustbebetween1and${MAX_NUM_VALIDATORS} (inclusive).`,); }
Figure5.3:Lengthcheckin updateAuthority (lisk-
sdk/framework/src/modules/poa/commands/update_authority.ts#33-37) hasheye
25LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

However,if minItems and maxItems fieldswereaddedtotheschemaforthiscommand (updateAuthoritySchema
),thischeckwouldbedoneautomaticallybytheschema validator. Recommendations Shortterm,addthesuggested
validator.validate checkto endpoint.ts andaddthe minItems and maxItems fieldstothe
updateAuthoritySchema . Longterm,ensurethatallLiskschemasareasexpressiveaspossible. hasheye
26LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

6.UpdateauthoritycommandallowsvalidatorstohaveOweight Severity:InformationalDifficulty:High
Type:DataValidationFindingID:TOB-LISK2-6 Target: lisk-
sdk/framework/src/modules/poa/commands/update_authority.ts Description
Whentheproofofauthoritymoduleinitializesitsgenesisstate,itchecksthatallactive
validatorshavenonzero(andpositive)weights: //Checkthattheweightpropertyofeveryentryinthe
snapshotSubstore.activeValidatorsarrayisapositiveinteger. if(activeValidators[i].weight≤BigInt(0))
{ thrownewError('`activeValidators`weightmustbepositiveinteger.'); }
Figure6.1:Validatorweightcheckingenesisinitialization (lisk-
sdk/framework/src/modules/poa/module.ts#247-250)

However,thesamecheckisnotmadebytheupdateauthoritycommand. Thismeansthatthecurrentvalidatorscouldvotetoapproveanewvalidatorlistthat includesvalidatorswith0weight.Wecouldnotfindanywaythatthiscouldbeexploitedto causeasecurityproblem.However,itwouldbeagoodpracticetoensurethattheinvariant “foreachi,activeValidators[i].weight>0” alwaysholds. Recommendations Shortterm,addacheckwhichthrowsanexceptionifavalidatorhas0weight. Longterm,ensurethatotherexpectedinvariantsonweightsareensured. hasheye 27LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

7.IncorrectMIN_SINT_32constant Severity:InformationalDifficulty:High Type:DataValidationFindingID:TOB-LISK2-7 Target: lisk-sdk/elements/lisk-validator/src/constants.ts Description The MIN_SINT_32 constant,usedbyLisk’sschemavalidator,isincorrect:itsvalueis -2147483647,butshouldbe-2147483648,sincethisistheminimumpossible32-bitsigned integer. ExploitScenario Alegitimateusertriestousethevalue-2147483648ina sint32 fieldofamessage,butis unabletobecausetheschemavalidatorrejectsthemessage. Recommendations Changethevalueofthe MIN_SINT_32 constantto-2147483648. hasheye 28LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

8.Eventerrorsarereverted Severity:LowDifficulty:Low Type:AuditingandLoggingFindingID:TOB-LISK2-8 Target: lisk-sdk/framework/src/modules/nft/events/{destroy.ts,lock.ts} Description The DestroyEvent andthe LockEvent classes’errormethodsdonotcallthe BaseEvent class’s add methodwiththe noRevert argumentsetto true .The noRevert argumentis usedtopreventrevertingspecificeventsevenwhencommandexecutionfails;itisusefulto emiterroreventsthatdescribetheerrorthatcausedtheexecutionfailure.Sincethe DestroyEvent andthe LockEvent classes’errormethodsdonotpassthisargumentto the add function,asshowninfigures8.1and8.2,anditsdefaultvalueis false ,theerror events—emittedwhenatransactionisabouttofail—willneverbeseenbytheuser. publicerror(ctx:EventQueuer,data:DestroyEventData,result:NftErrorEventResult): void{ this.add(ctx, {...data,result},[data.address,data.nftID]); } Figure8.1:TheerrormethodoftheDestroyEventclass (lisk-sdk/framework/src/modules/nft/events/destroy.ts#56-58) publicerror(ctx:EventQueuer,data:LockEventData,result:NftErrorEventResult){ this.add(ctx, {...data,result},[Buffer.from(data.module),data.nftID]); } Figure8.2:The error methodofthe LockEvent class (lisk-sdk/framework/src/modules/nft/events/lock.ts#63-65) WefoundthisissuewiththeSemgreprule error_event_is_not_revert ,whichwe providedinthefirstphaseofthisaudit. rules: -id:error_event_is_not_revert message:Aneventclasshasanerrorfunctionthataddsaneventthatisnot persisted. languages:[typescript] severity:WARNING patterns: -pattern-inside:> hasheye 29LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

class$CLASSextendsBaseEvent{ ... } -pattern:> error(...){ ... } -pattern-not:> error(...){ ... this.add(...,true); ... } Figure8.3:The error_event_is_not_revert Semgreprule ExploitScenario1 Anattackerhasanattackagainstacustomsidechainthatinvolvesbruteforcingthe destroy methodoftheNFTmodule.Noerroreventisemittedforthesefailures.The attackgoesunnoticed. ExploitScenario2 Auserorblockchaindevelopertriestodestroyan NFT.Theircalltothe destroy method failsbutnoerroreventisemitted.Theuserfailsortakesaverylongtimetodebugtheroot causeofthefailure. Recommendations Shortterm,onevery error methodofasubclassofthe BaseEvent class(specificallyfor DestroyEvent and LockEvent),passthe noRevert argumentsetto true .Thiswill ensurethaterroreventsarenotrevertedwhencommandsfail. Longterm,toavoidsimilarerrorsinthefuture,integratethe error_event_is_not_revert Semgreprule,aswellaseverySemgreprulethatwe providedinthefirstphaseoftheaudit,intotheCI/CDpipeline.Additionally,consider addingan error methodtothe BaseEvent classthatissimilartothe add method,but withthe noRevert argumentdefaultingto true .Thiswillmakeiteasierfordeveloperstowritecodewithoutthisbuggypattern. hasheye 30LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

9.UnspecifiedandpoorlynamedNFTendpoints Severity:InformationalDifficulty:High Type:ConfigurationFindingID:TOB-LISK2-9 Target:NFTmodule Description The getCollectionIDs , collectionExists ,and isNFTSupported endpointsofthe NFTmoduleexistinthecodebutarenotdefinedinLIP-0052. Furthermore,the getCollectionIDs and collectionExists functionsincorrectly performacheckifanNFTissupported.Insteadofusingthe isNFTSupported methodof theNFTmodule,thesefunctionsusethecodeshowninfigure9.1. constsupportedNFTsStore=this.stores.get(SupportedNFTsStore); constchainExists=awaitsupportedNFTsStore.has(ctx,chainID); Figure9.1: lisk-sdk/framework/src/modules/nft/endpoint.ts#180-182 Thischeckdoesnottakeintoaccounttheexistenceofthe ALL_SUPPORTED_NFTS_KEY key inthe SupportedNFTsStore .Also, getCollectionIDs returnsthesamevalueifthere arenosupportedcollectionsorifallcollectionsaresupportedforagivenchain. Finally,thefunction’snamesarenotclear.Giventheseunclearnamesandnonexistent

specification,wearenotentirelysureofthefunctions’intendedfunctionality,but alternativenamescouldbe getSupportedCollectionIDs and isCollectionIDSupported . Recommendations Shortterm,specifythefunctionsdescribedaboveintheLIP,renamethemwithaclearer name,andfixtheirfunctionalityaccordinglywiththedevelopedspecification. Longterm,createaprocesstolimitaddingcodetomoduleswithoutitbeingspecifiedina LIP. hasheye 31LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

10.TheremoveSupportAllNFTsfunctionmaynotremovesupportforallNFTs Severity:MediumDifficulty:Low Type:DataValidationFindingID:TOB-LISK2-10 Target: lisk-sdk/framework/src/modules/nft/method.ts Description The removeSupportAllNFTs methodoftheNFTmodulemaynotremovesupportofall NFTsifthe ALL_SUPPORTED_NFTS_KEY ispresent.The removeSupportAllNFTs method (figure10.1)removesallkeysinthe SupportedNFTsStore thatarereturnedbythe supportedNFTsStore.getAll method(figure10.2).However,the getAll functionwill neverreturnthe ALL_SUPPORTED_NFTS_KEY keybecauseititeratesoverallkeysofsize LENGTH_CHAIN_ID but ALL_SUPPORTED_NFTS_KEY istheemptystring(i.e.,doesnothave alengthof LENGTH_CHAIN_ID). publicasyncremoveSupportAllNFTs(methodContext:MethodContext):Promise<void>{ constsupportedNFTsStore=this.stores.get(SupportedNFTsStore); constallSupportedNFTs=awaitsupportedNFTsStore.getAll(methodContext); for(const{key}ofallSupportedNFTs){ awaitsupportedNFTsStore.del(methodContext,key); } this.events.get(AllNFTsSupportRemovedEvent).log(methodContext); } Figure10.1: lisk-sdk/framework/src/modules/nft/method.ts#709-719 publicasyncgetAll(context:ImmutableStoreGetter,):Promise<{key:Buffer;value:SupportedNFTsStoreData}[]>{ returnthis.iterate(context,{ gte:Buffer.alloc(LENGTH_CHAIN_ID,0), lte:Buffer.alloc(LENGTH_CHAIN_ID,255), }); } Figure10.2: lisk-sdk/framework/src/modules/nft/stores/supported_nfts.ts#63-70 Weusedthetestfromfigure10.3toconfirmtheissuedescribedabove. it('shouldremoveallexistingentriesevenifthe ALL_SUPPORTED_NFTS_KEY hasheye 32LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

entryexists',async()⇒{ awaitsupportedNFTsStore.save(methodContext, ALL_SUPPORTED_NFTS_KEY,{ supportedCollectionIDArray:[], }); await expect(method.removeSupportAllNFTs(methodContext)).resolves.toBeUndefined(); awaitexpect(supportedNFTsStore.has(methodContext, ALL_SUPPORTED_NFTS_KEY)).resolves.toBeFalse(); checkEventResult(methodContext.eventQueue,1, AllNFTsSupportRemovedEvent,0,{},null); }); Figure10.3:Testthatfailsbecausethe removeSupportAllNFTs doesnotdeletethe ALL_SUPPORTED_NFTS_KEY key.Highlightedinredisthecheckthatfails. ExploitScenario Amodulereliesonthe removeSupportAllNFTs functionforremovingsupportforall NFTsunderspecificconditions.Thecallto removeSupportAllNFTs failstoremove supportforallNFTs.Anattackerexploitsthisfacttobreakthemodule’sassumptionsand exploitthesidechain. Recommendations Shortterm,inthe removeSupportAllNFTs function,alsoremovethe ALL_SUPPORTED_NFTS_KEY key.Addthetestinfigure10.3totheexistingtestsuite. Longterm,writemorecomplextestsfor removeSupportAllNFTs andsimilarfunctions. Edgecasesandmagicvaluessuchas ALL_SUPPORTED_NFTS_KEY shouldbethoroughly tested. hasheye 33LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

11.BuginremoveSupportAllNFTstests Severity:InformationalDifficulty:High Type:TestingFindingID:TOB-LISK2-11 Target: lisk-sdk/framework/test/unit/modules/nft/method.spec.ts Description Thetestforthe removeSupportAllNFTs functionincorrectlysavesarandomwrongvalue tothe supportedNFTsStore store. Thetestshouldworkby:1)creatingarandom chainID ,2)savingthat chainID tothe supportedNFTsStore store,3)removingthesupportforthat chainID ,and4)checking thatthe supportedNFTsStore nolongercontainsthe chainID key.Thebugisinstep2, wherethecodeisjustsavingarandomchainIDinsteadofusingthechainIDvariable (highlightedinredinfigure11.1). constchainID=utils.getRandomBytes(LENGTH_CHAIN_ID); awaitsupportedNFTsStore.save(methodContext,utils.getRandomBytes(LENGTH_CHAIN_ID), { supportedCollectionIDArray:[], }); awaitexpect(method.removeSupportAllNFTs(methodContext)).resolves.toBeUndefined(); awaitexpect(supportedNFTsStore.has(methodContext,chainID)).resolves.toBeFalse(); Figure11.1: lisk-sdk/framework/test/unit/modules/nft/method.spec.ts#1183-1191 Furthermore,byremovingthecallto supportedNFTsStore.save altogether,thetessttill passes,whichisundesirable.ThecodeshouldfirstcheckthattheNFTissupported,remove supportforallNFTs,andcheckiftheNFTisnolongersupported. Recommendations Shortterm,fixthetestdescribedabovebyreplacingthecallto supportedNFTsStore.sav e’s utils.getRandomBytes(LENGTH_CHAIN_ID) argument withthe chainID variable.Also,addacheckinthetestthatensuresthattheNFTis supportedbeforeremovingit. Longterm,useatoolsuchasnecessisttofindotherissuesintests. hasheye 34LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

12.BouncedNFTtransfersmaycauseeventstocontainincorrectdata Severity:LowDifficulty:Low Type:AuditingandLoggingFindingID:TOB-LISK2-12 Target: lisk-

sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts Description Whenanerroroccursonacross-chainNFTtransferandthestatusoftheCCMisnot CCM_STATUS_CODE_OK ,the recipientAddress variableissettothevalueofthe senderAddress variableintwolocations:hereandhere.The senderAddress variableis notupdated. Then,attheendofthefunction,a CcmTransferEvent eventisemittedwiththe senderAddress and recipientAddress values(figure12.1),which,asexplainedabove, willalwayshavethesamevalueonstatusdifferentfrom CCM_STATUS_CODE_OK .Thesame problemexistsinthetokenmodule. this.events.get(CcmTransferEvent).log(context,{ senderAddress, recipientAddress, nftID, }); Figure12.1: lisk-sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts#157-161 Anotherproblemisthelackofsendingandreceiving chainID dataintheevent.Whilethe tokenmoduleincludesthereceiving chainID ,theNFTmoduleincludesneither(asshown infigure12.1). ExploitScenario AdeveloperistracingwhereanNFTwastransferredtoandfromtoauditanongoing attack.Inthelogs,thedeveloperseesthattheNFTwastransferredfromaccountAto accountBandthen,becauseanerroroccurred,thelogwillshowatransferfromaccountA toaccountA.Inbothcases,thereceivingandsending chainID sarenotincludedinthe eventdata.Thisconfusesthedeveloperandslowsdowntheauditingprocess. Recommendations Shortterm,inboththeNFTandtokenmodule’s CcmTransferEvent event,updatethe senderAddress tocorrectlyshowthesenderaddressoftheNFTortokenamount. Considerifftheeventdatashouldalsoincludethesendingandreceiving chainID s,and hasheye 35LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

updatebothmodulesaccordingly.Thiswillmakethe CcmTransferEvent eventsemitted bytheNFTandTokenmodulesmoreexpressiveandconsistentwitheachother. Longterm,reviewtheeventsemittedinallothercommands.Ensurethattheyare consistentwithsimilarcommandsandthattheyaresufficienttotracetheownershipof assetsandotherrequiredproperties. hasheye 36LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

13.AnNFT'sattributesArraycanhaveduplicatemodules Severity:LowDifficulty:Low Type:DataValidationFindingID:TOB-LISK2-13 Target:NFTmodule Description TheNFTmoduledefinesthe createNFTEntry function,whichisresponsibleforcreatinga newNFTandensuringthattheNFT’s attributesArray arraydoesnothaveduplicate modules,ashighlightedinfigure13.1. publicasynccreateNFTEntry(methodContext:MethodContext, address:Buffer, nftID:Buffer, attributesArray:NFTAttributes[],):Promise<void>{ constmoduleNames=[]; for(constitemofattributesArray){ moduleNames.push(item.module); } if(newSet(moduleNames).size≠=attributesArray.length){ thrownewError('Invalidattributesarrayprovided'); } constnftStore=this.stores.get(NFTStore); awaitnftStore.save(methodContext,nftID,{ owner:address, attributesArray, }); } Figure13.1: lisk-sdk/framework/src/modules/nft/internal_method.ts#63-83 The createNFTEntry functionisnevercalled;instead,the nftStore.save methodis calleddirectlyinmanylocations.Insomelocations,suchasinthe create method,the attributeArray uniquenesscheckiscopy-pasted,whileinothersthisuniqueness propertyisnotenforced. Inparticular,twoplacesmayresultintheNFThavingduplicatemoduleattributes: • WhenaforeignNFTisreceived (lisk-sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts#L142-L145) hasheye 37LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

• WhenaforeignNFTisbounced (lisk-sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts#L149-L152) Inthe recover functionandcross-chaintransferstotheNFT’snativechain,thisproblem alsoexistsbutisnotexploitablebecausetheincomingNFTattributesaredroppedbythe currentimplementationof getNewAttributes : • lisk-sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts#L116 • lisk-sdk/framework/src/modules/nft/method.ts#L977 The createUserEntry functionisalsonotusedhereandhere,andthe createEscrowEntry functionisnotusedhere. ExploitScenario AnattackersendsanNFTwithduplicateattributesfromchainA(theNFT’snativechain)to chainB.AcustommoduleofchainBperformsvalidationoftheNFT’sattributesbyiterating overtheattributesarrayuntilitfindsthefirstinstancecorrespondingtomoduleX(e.g., usingthearray’s find method)andthendoingthecheck.Later,thesamecustommodule getstheattributesformoduleXinadifferentwaythatreturnsthelastinstanceonthe attributesarrayinsteadofthefirst(e.g.,byiteratingoverthewholearrayandaddingthe valuestoamapforfasteraccess).Onlythefirstinstanceischecked,breakingthe guaranteesthatthemodulehasfortheattributesandpotentiallyleadingtoasecurity problem. Recommendations Shortterm,ineverylocationwhereanNFTiscreatedorupdated,useapurposelybuilt function(e.g.,the createNFTEntry function)tocreatetheNFTandensurethecorrectness ofitsproperties.Avoidcalling nftStore.save andsimilarmethodsinothercode locations. Longterm,ingeneral,avoidmodifyingthestore’sstatedirectly,especiallywhenspecific propertiesneedtobeenforced.Instead,createpurposelybuiltfunctionsthatenforcethe propertiesinasinglecentralizedlocationthatiseasytoauditandavoidscoderepetition.

Seefigure4.4fromfindingTOB-LISK2-4toseewhatthefinalresultcouldlooklike. hasheye
38LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

14.BouncedmessagesmaybeabusedtobypassNFTfees Severity:MediumDifficulty:Medium
Type:DataValidationFindingID:TOB-LISK2-14 Target: lisk-
sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts Description OnCCMswithastatusdifferentfrom
CCM_STATUS_CODE_OK (figure14.1highlightedin yellow),theNFTcreationdoesnotincurafee.
if(status===CCM_STATUS_CODE_OK){ this._feeMethod.payFee(getMethodContext(),BigInt(FEE_CREATE_NFT));
awaitnftStore.save(getMethodContext(),nftID,{ owner:recipientAddress,
attributesArray:receivedAttributesasNFTAttributes[], });
awaitthis._internalMethod.createUserEntry(ctx,recipientAddress,nftID); }else{
recipientAddress=senderAddress; awaitnftStore.save(getMethodContext(),nftID,{
owner:recipientAddress, attributesArray:receivedAttributesasNFTAttributes[], });
awaitthis._internalMethod.createUserEntry(ctx,recipientAddress,nftID); } Figure14.1: lisk-
sdk/framework/src/modules/nft/cc_commands/cc_transfer.ts#140-154
Thiscodeworksthiswaybecausemessageswithastatusdifferentfrom CCM_STATUS_CODE_OK
areexpectedtobeabouncedCCM,i.e.,aCCMwheretheNFTis beingrecreatedafterapreviouscross-
chaintransferfailed.However,thereisnothing
preventingamaliciousidechainfrominitiatingaCCMwithanystatus,enablingthemto transferNFTscross-
chainwithoutpayingthefee. ExploitScenario
ChainAsendsanNFT(nativetoChainA)tochainB,settingthemessagestatustoavalue differentfrom
CCM_STATUS_CODE_OK (e.g., CCMStatusCode.MODULE_NOT_SUPPORTED).
ChainBreceivestheforeignNFTandcreatestheNFTentrywithoutthefeepayment. hasheye
39LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Recommendations Shortterm,enforceafeepaymentevenonbouncedmessages.Ifthisisnotpossibleor
wanted,modifytheinteroperabilitymoduletoguaranteethatthemainchainenforcesthat
bouncedmessagescannotbeinitiatedbymaliciousidechains.
Longterm,reviewothermodules’handlingofbouncedmessagestoensuresimilar problemsdonotexist. hasheye
40LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

15.NFTrecoverfunctionmaycrash Severity:InformationalDifficulty:Medium
Type:DataValidationFindingID:TOB-LISK2-15 Target: lisk-sdk/framework/src/modules/nft/method.ts
Description The recover functionoftheNFTmoduledoesnotverifythattheNFTtoberecovered existsinits
NFTStore .Thiscausesthe nftStore.get callhighlightedinfigure15.1to crash.
constnftData=awaitnftStore.get(methodContext,nftID); if(!nftData.owner.equals(terminatedChainID)){
Figure15.1: lisk-sdk/framework/src/modules/nft/method.ts#940-941 ExploitScenario
AmaliciousidechainsendsafakerecoveredNFTthatdoesnotexistonitsnativechain. Thenativechain’s recover
functioncrashes. Recommendations
Shortterm,addachecktoensurethattheNFTexistsbeforeaccessingitsdata.Thiswill ensurethe recover
functionthrowsanexceptionexplicitlyinsteadofcrashing unexpectedly.
Longterm,writefuzzingharnessesthatsendarbitraryobjects(thatarevalidaccordingto givenschema)tothe
recover methodtosimulatethebehaviorofmaliciousidechains. hasheye
41LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

16.NFTrecoverfunctiondoesproperlyvalidateNFTattributesarray Severity:InformationalDifficulty:Medium
Type:DataValidationFindingID:TOB-LISK2-16 Target: lisk-sdk/framework/src/modules/nft/method.ts
Description The recover functionoftheNFTmoduledoesnotcall validator.validate onthe storeValue
variableafterdecodingthevalue(figure16.1). try{ decodedValue=codec.decode<NFTStoreData>
(nftStoreSchema,storeValue); }catch(error){ isDecodable=false; } Figure16.1: lisk-
sdk/framework/src/modules/nft/method.ts#902-908
Assuch,theNFTattributes’modulenamesarenotvalidatedforlengthorpatternwiththe nftStoreSchema
,whichisshowninfigure16.2. exportconstnftStoreSchema={ $id:'/nft/store/nft', type:'object',
required:['owner','attributesArray'], properties:{ owner:{ dataType:'bytes', fieldNumber:1, },
attributesArray:{ type:'array', fieldNumber:2, items:{ type:'object', required:
['module','attributes'], properties:{ module:{ dataType:'string', minLength:MIN_LENGTH_MODULE_NAME,
maxLength:MAX_LENGTH_MODULE_NAME, pattern:'^[a-zA-Z0-9]*$', fieldNumber:1, }, hasheye
42LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

attributes:{ dataType:'bytes', fieldNumber:2, }, }, }, }, }, }; Figure16.2: lisk-
sdk/framework/src/modules/nft/stores/nft.ts#28-59 Thefollowingtestcanbeusedtoconfirmthatthecallto
codec.decode doesnotvalidate astring’slength. describe('length_validation',()=>{
constschema_with_max_length={ $id:'/lisk/decode/with_max_length', type:'object', properties:{ foo:{
dataType:'string', fieldNumber:1, minLength:2, maxLength:2, }, }, };
it('shouldworkwithavalidlengthof2',()=>{ constres=codec.encode(schema_with_max_length,{foo:"12"});
expect(codec.decode(schema_with_max_length,res)).toEqual({foo:"12"}); });
it('shouldnotworkwithlengthdifferentthan2',()=>{ constres=codec.encode(schema_with_max_length,

{foo:"12345"}); expect(codec.decode(schema_with_max_length,res)).toThrow();//Thisdoes notthrow }); }); Figure16.3:Testthatconfirmsthatcallsto codec.decode donotvalidateastring’slength Thisfindingisinformationalbecausetheattributesarenotusedunlessacustommodule implementsthe getNewAttributes function. ExploitScenario AnattackerrecoversanNFTfromachaintheyowntoasidechainthatreimplementsthe getNewAttributes functiontousethereceivedNFTattributes.The NFTStore storesaves datathatisinconsistentwiththeschema. hasheye 43LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Recommendations Shortterm,havethecodecallthe validator.validate functiononthedecoded NFTStoreData data. Longterm,revieweveryothermodule’s recover methodsforthesameissue. hasheye 44LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

17.Thecodec.encodefunctionencodeslargerthanexpectedintegers Severity:LowDifficulty:High Type:DataValidationFindingID:TOB-LISK2-17 Target: Codec class Description The encode functionofthe Codec classsuccessfullyencodesintegerslargerthan expected.Thisissuecanbereplicatedwiththetestshowninfigure17.1,inwhichavalueof 2 100 issuccessfullyencodedasa uint64 (whilethemaximumvalueshouldbe2 64 -1). describe('uint_validation', ()⇒{ constschema_uint64={ $id:'test/uint_validation', type:'object', required:['amount'], properties:{ amount:{ dataType:'uint64', fieldNumber:1, }, }, }; it('uint64encodingwithlargerthanexpectedvalues',()⇒{ expect(codec.encode(schema_uint64,{amount: BigInt(2)**BigInt(100)})).toThrow();//Thisdoesnotthrow }); }); Figure17.1:Testthatshowstheencodefunctionofthe Codec classsuccessfullyencoding integerslargerthanexpected Callingthe decode methodofthe Codec classonthisencodeddatafails,breakingthe round-trippropertyofthe encode and decode functions. ExploitScenario Ausersuccessfullystoresavalueinamodule’sstorethatislargerthanthemaximum integer.Theuserattemptstousethemoduleagainbutisunabletobecausethecallto decode fails. hasheye 45LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Recommendations Shortterm,addchecksinthecodetopreventintegerslargerthanthemaximumvalueor smallerthantheminimumvaluecannotbeencodedordecoded. Longterm,extendthetestsuiteforintegerencodinganddecodingtodetecttheseand similarissues.Alwaysteststhecornercases,suchasthemaximumandminimuminteger boundaries. hasheye 46LiskSDKv6.1Sapphire,NFTandPoAAssessment 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 47LiskSDKv6.1Sapphire,NFTandPoAAssessment 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 48LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

B.CodeMaturityCategories Thefollowingtablesdescribethecodematuritycategoriesandratingcriteriausedinthis document.

CodeMaturityCategories CategoryDescription
ArithmeticTheproperuseofmathematicaloperationsandsemantics
AuditingTheuseofeventauditingandloggingtosupportmonitoring Authentication/ AccessControls
Theuseofrobustaccesscontrolstohandleidentificationand
authorizationandtoensuresafeinteractionswiththesystem Complexity Management
Thepresenceofclearstructuresdesignedtomanagesystemcomplexity,
includingtheseparationofsystemlogicintoclearlydefinedfunctions 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.
WeakManyissuesthataffectsystemsafetywerefound.
MissingArequiredcomponentismissing,significantlyaffectingsystemsafety. hasheye
49LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

NotApplicableThecategoryisnotapplicabletothisreview.
NotConsideredThecategorywasnotconsideredinthisreview. Further Investigation Required
Furtherinvestigationisrequiredtoreachameaningfulconclusion. hasheye
50LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

C.CodeQualityRecommendations Thisappendixcontainsfindingsthatdonothaveimmediateorobvioussecurity
implications.However,theymayfacilitateexploitchainstargetingothervulnerabilities,may
becomeeasilyexploitableinfuturereleases,ormaydecreasecodereadability.We
recommendfixingthefollowingissues: NFTModule
1.IncorrectregexinLIP.TheLIP'suserStoreSchemaschemaspecificationdefinesthe patternforthe
lockingModule as ^[a-zA-Z0-9]*$] .Thelast] isatypoand shouldnotbepresent.Thepatternshouldbe ^[a-
zA-Z0-9]*$ .Thecorresponding implementationiscorrect. 2.Constantsaredefinedbuthard-
codedversionsareused.In module.ts#L124-L126 , 'nft' isusedinsteadof MODULE_NAME_NFT .
3.Unnecessarycallsto validator.validate .IntheNFTmodule’stransferand
transfercrosschaincommands,thecodecalls validator.validate tovalidate
thecommands’parameters.Calling validator.validate onacommand's verify
functionisunnecessarybecausethischeckisalreadydoneinthestate machinelogic. 4.Useofdeprecated
Buffer.slice function.Thecodesuses Buffer.slice —a deprecatedfunction—in method.ts#L77 ,
method.ts#L196 ,and method.ts#L289 .Instead,thecodeshoulduse Buffer.subarray .
5.CheckmissingfromLIP.The getChainID functionimplementationhasacheckfor the nftID
length,whichisnotinthecorrespondingLIPspecification.TheLIPshould beupdatedtoincludethischeck.
6.BadpseudocodenotationintheLIP.TheLIPcontainsinconsistentnotationina coupleoflocations:
a.Thespecificationofthe getCollectionID functionhasapoornotationfor sub-arrayingthe nftID
variable.TheLIPuses nftID[`LENGTH_CHAIN_ID`:(`LENGTH_CHAIN_ID`+ `LENGTH_COLLECTION_ID`)]
.Werecommendremovingthe ` characters, resultinginthisnotation: nftID[LENGTH_CHAIN_ID:
(LENGTH_CHAIN_ID+ LENGTH_COLLECTION_ID)] . b.Thespecificationofthe getAttributes
functionhasareturnstatement withunnecessaryparenthesis.TheLIP’sreturnstatementisrepresentedas
return(entry['attributes']) ,buttheparenthesisareunnecessary. hasheye
51LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

Instead,tokeepthenotationconsistentwithotherreturnsintheLIP,the pseudo-codeshouldbe
returnentry['attributes'] . 7.Lackofconstantdefinition.Thecodecalculatesthelengthofthe index
portionof an nftID intwolocations(method.ts#L276 ,and method.ts#L315)withthecode
indexLength=LENGTH_NFT_ID-LENGTH_CHAIN_ID- LENGTH_COLLECTION_ID
.Instead,thisvalueshouldbedefinedasaconstantlike LENGTH_CHAIN_ID ,and LENGTH_COLLECTION_ID .
8.Inconsistenterrorhandlingincommand verify functions.Forexample,the
NFTmodulethrowsanexceptiononerrors,whilethetokenmodulereturnsa VerificationResult witha
VerifyStatus of TransactionVerifyResult.INVALID .Inbothcases,thestatemachinehandles
theerrorssimilarly(onlyalteringthedebuglogthatiscreated);however,theerror handlingforallcommand’s
verify functionsshouldbeconsistent. 9.Inconsistenteventdata.The CreateEvent eventdataincludesthe
collectionID field.Thisfieldisnotpresentinthe lock , unlock ,or destroy events,whichincludeonlythe
nftID (fromwhichtheNFT’scollectionIDcanbe
extracted).Alloftheseeventsthatmodifythestateof NFTsshouldbeconsistentin
thedatatheyinclude,eitherincludingthe collectionID fieldinallofthemorin noneofthem. 10.CheckinLIP-

0052butnotintheimplementation.Wefoundthefollowing instanceswherechecksarenotdefinedinthespecificationbutarepresentinthe implementation: a.Inthe removeSupportAllNFTsFromCollection function,the chainID.equals(this._config.ownChainID) checkisnotdefinedinthe LIPspecification. b.Inthe supportAllNFTs function,thecheckthattestsifthe ALL_SUPPORTED_NFTS_KEYkeyispresentintheSupportedNFTsStoreis missingfromtheLIP. PoAModule 1. Misspelled,unusedconstant.In poa/constants.ts ,both LENGTH_PROOF_OF_POSSESSION and LENGTH_PROOF_OF_POSESSION aredefined, bothequaling96. LENGTH_PROOF_OF_POSESSION isneverused. 2. Redundant,unusedtype.In poa/stores/snapshot.ts ,the Validator typeis definedbutneverused.Itisidenticaltothe ActiveValidator type,importedfrom poa/types.ts . ActiveValidator isusedlaterin poa/stores/snapshot.ts . hasheye 52LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

3. Hard-coded SUBSTORE_PREFIX constants.TheLIPspecifyingthePoAmodule specifiesfour SUBSTORE_PREFIX constants.Theseconstantsarehardcodedinthe implementation(in poa/module.ts ,intheconstructorof PoAModule),ratherthan beingdefinedasconstantsin poa/constants.ts . 4. Constantsaredefinedbuthard-codedversionsareused: a.In poa/module.ts ,line88, 'poa' isusedinsteadof MODULE_NAME_POA . b.In poa/module.ts ,line216, /^[a-z0-9!@$&_.]+$/g isusedinsteadof POA_VALIDATOR_NAME_REGEX .Whilethehard-codedvalueisslightly differentfromtheconstant(thehard-codedvalueincludesa g forglobalat theend,whiletheconstantdoesnot),theeffectwillbethesame. c.In poa/endpoint.ts ,onlines73and74, 20 isusedinsteadof NUM_BYTES_ADDRESS . 5. InconsistentuseofBuffer.from.Somecodelocationsuse Buffer.from(str, 'utf-8') whileotherlocationsuse Buffer.from(str) ,leavingoutthe 'utf-8' argument.Theeffectisthesame,since 'utf-8' isthedefaultvalueofthat argument. 6. RegisterAuthorityParams fieldsareordereddifferentlythanintheLIP.The generatorKey and proofOfPossession fieldsareswapped.Thiscouldpotentially causeschemamatchingproblemsifJSONserializationisdoneinthewrongorder. However,wedidnotfindanysecurityproblemsrelatedtothisissue. 7. Unnecessarycallsto validator.validate .InthePoAmodule’sregister authorityandupdategeneratorkeycommands,thecodecalls validator.validate tovalidatethecommandsparameters.Calling validator.validate onacommand’s verify functionisunnecessarybecause thischeckisalreadydoneinthestatemachinelogic. 8. Arrayorderingischeckedmanuallywhenahelperfunctioncouldbeused.In the PoAModule.initGenesisState function,thereisacheckforwhetherthe validatorAddresses listissorted,bycomparingeachofitselementsagainstthe correspondingelementin [...validatorAddresses].sort() .Instead,the objectUtils.isBufferArrayOrdered helperfunctionshouldbeused.Thesame appliestothecheckdonelaterinthefunctiononthe activeValidators list: objectUtils.isBufferArrayOrdered(activeValidatorAddresses) should beused. 9. The shuffleValidatorList functionisdefinedbothinPoAandPoS.Bothdo essentiallythesamething,butareimplementedinslightlydifferentways.Ideally, theyshoulduseacommonimplementation. hasheye 53LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

10. InLIP-0047forPoA, genesisPoAStoreSchema.snapshotSubstore ismissinga fieldnumber.ThisproblemoccursonlyintheLIP,andnotintheimplementation. 11. The getAllValidatorsResponseSchema schemaspecifiesthatweightsshould havea 'uint64' format,whiletheimplementationmayassignthemtobe blankstrings.Thisisnotasecurityissue,sincethereisnoformaldefinitionofthe 'uint64' formatanyway(seethefixreviewfor TOB-LISK-52 fromthefirstphase oftheaudit),andsincetheresponseisnevervalidatedagainsttheschema. 12. Unnecessarystorelookup.Inthe PoAEndpoint.getAllValidators function, validatorStore.iterate isusedtogetalistofvalidator (address,name) pairs.However,lateron,theaddressesinthereturnedlistareagainlookedupusing validatorStore.get(context,data.key) ,when data.value couldbeused instead. 13. Unnecessarycallto getAddressFromPublicKey .Theregisterauthority commanduses address.getAddressFromPublicKey(context.transaction.senderPublicK ey) inits verify functiontogetthetransaction’ssenderaddress,whenitcould insteaduse context.transaction.senderAddress . hasheye 54LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

D.StaticAnalysisToolConfiguration

AspartofthisassessmentWeperformedautomatedstatictestingusingSemgrepand CodeQL. Semgrep WeperformedstaticanalysisontheLiskSDKNFTandPoAmodulesusingthefollowing publicrulesets: • p/javascript • p/r2c • p/r2c-security-audit • p/r2c-best-practices • p/nodejs • p/nodejsscan Wealsousedthecustomrulesprovidedinthepreviouspartofthisaudit: • error_event_is_not_revert • get_modify_no_set_on_stores • module_registration_of_correct_class • module_stores_same_index • schema_min_max_items_without_array • schema_property_element_without_field_number • schema_with_duplicate_field_number • schema_with_field_not_required • verify_without_schema_verify • schema_with_required_that_is_not_a_property • schema_int_format_with_integer_type • schema_hardcoded_pattern hasheye 55LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

CodeQL WeusedCodeQLtoanalyzetheLiskcodebases.Weusedourprivate tob-javascript-all querysuite,whichincludespublicJavaScriptqueriesandsome privatequeries.FigureD.1showsthecommandsusedtocreatetheCodeQLdatabaseand runthequeries. #Createthejavascriptdatabase codeqldatabasecreatecodeql.db--language=javascript #Runalljavascriptqueries codeqldatabaseanalyzecodeql.db--format=sarif-latest--output=codeql_res.sarif-- tob-javascript-all.qls FigureD.1:CommandsusedtorunCodeQL hasheye 56LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

E.FixReviewResults Whenundertakingafixreview,hasheyereviewsthefixesimplementedforissues identifiedintheoriginalreport.Thisworkinvolvesareviewofspecificareasofthesource codeandsystemconfiguration,notcomprehensiveanalysisofthesystem. FromOctober18toOctober25,hasheyereviewedthefixesandmitigations implementedbytheLiskFoundationteamfortheissuesidentifiedinthisreport.We reviewedeachfixtodetermineitseffectivenessinresolvingtheassociatedissue. Insummary,theLiskFoundationteamhasresolvedallissuesdescribedinthisreport.For additionalinformation,pleaseseetheDetailedFixReviewResultsbelow. IDTitleStatus 1Impossibletosendcross-chainNFTtransfersincertainconfigurationsResolved 2NFTmodulehasrepeatedtransferfunctionalityResolved 3NFTlockmethoddoesnotcheckifmoduleisNFT_NOT_LOCKEDResolved 4NFTmethodsAPIcouldbeimprovedResolved 5ChecksdonemanuallyinsteadofusingschemavalidationResolved 6Updateauthoritycommandallowsvalidatorstohave0weightResolved 7IncorrectMIN_SINT_32constantResolved 8EventerrorsarerevertedResolved 9UnspecifiedandpoorlynamedNFTendpointsResolved hasheye 57LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

10TheremoveSupportAllNFTsfunctionmaynotremovesupportforall NFTs Resolved 11BuginremoveSupportAllNFTstestsResolved 12BouncedNFTtransfersmaycauseeventstocontainincorrectdataResolved 13AnNFT'sattributesArraycanhaveduplicatemodulesResolved 14BouncedmessagesmaybeabusedtobypassNFTfeesResolved 15NFTrecoverfunctionmaycrashResolved 16NFTrecoverfunctiondoesproperlyvalidateNFTattributesarrayResolved 17Thecodec.encodefunctionencodeslargerthanexpectedintegersResolved DetailedFixReviewResults TOB-LISK2-1:Impossibletosendcross-chainNFTtransfersincertainconfigurations ResolvedinPR#8963. TOB-LISK2-2:NFTmodulehasrepeatedtransferfunctionality ResolvedinPR#9005. TOB-LISK2-3:NFTlockmethoddoesnotcheckifmoduleisNFT_NOT_LOCKED ResolvedinPR#8992. TOB-LISK2-4:NFTmethodsAPIcouldbeimproved ResolvedinPRs#9034and#9005. TOB-LISK2-5:Checksdonemanuallyinsteadofusingschemavalidation ResolvedinPR#8967and#8953. TOB-LISK2-6:Updateauthoritycommandallowsvalidatorstohave0weight ResolvedinPR#8953. hasheye 58LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

TOB-LISK2-7:IncorrectMIN_SINT_32constant ResolvedinPR#8976. TOB-LISK2-8:Eventerrorsarereverted ResolvedinPR#8978. TOB-LISK2-9:UnspecifiedandpoorlynamedNFTendpoints ResolvedinPR#9036. TOB-LISK2-10:TheremoveSupportAllNFTsfunctionmaynotremovesupportforall NFTs ResolvedinPR#9017. TOB-LISK2-11:BuginremoveSupportAllNFTstests ResolvedinPR#9018. TOB-LISK2-12:BouncedNFTtransfersmaycauseeventstocontainincorrectdata ResolvedinPR#9011.WeencouragetheLiskteamtoapplythesamefixtothetoken module.Currently,thetokenmodulelogs receivingChainID butnot sendingChainID . TOB-LISK2-13:AnNFT'sattributesArraycanhaveduplicatemodules ResolvedinPR#9014. TOB-LISK2-14:BouncedmessagesmaybeabusedtobypassNFTfees ResolvedinPR#9011. TOB-LISK2-15:NFTrecoverfunctionmaycrash ResolvedinPR#9028. TOB-LISK2-16:NFTrecoverfunctiondoesproperlyvalidateNFTattributesarray ResolvedinPR#9000. TOB-LISK2-17:Thecodec.encodefunctionencodeslargerthanexpectedintegers ResolvedinPR#9010. hasheye 59LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC

F.FixReviewStatusCategories Thefollowingtabledescribesthestatusesusedtoindicatewhetheranissuehasbeen sufficientlyaddressed. FixStatus StatusDescription UndeterminedThestatusoftheissuewasnotdeterminedduringthisengagement. UnresolvedTheissuepersistsandhasnotbeenresolved. PartiallyResolvedTheissuepersistsbuthasbeenpartiallyresolved. ResolvedTheissuehasbeensufficientlyresolved. hasheye 60LiskSDKv6.1Sapphire,NFTandPoAAssessment PUBLIC