HashEye

// SMART CONTRACT SECURITY AUDIT

Prysm

Security assessment by HashEye · prepared for A private client

HASHEYE AUDITED

PROJECT	Prysm
CLIENT	A private client
CATEGORY	Blockchain
PUBLISHED	April 1, 2023
REPORT ID	research-prysm-2023-04-01-1ona3i

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-prysm-2023-04-01-1ona3i.

Prysm SecurityAssessment October23,2023 Preparedfor: PrivateClient
Preparedby:BenjaminSamuels,DominikCzarnota,andDamilolaEdwards

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 1Prysmv3.2.2SecurityAssessment PUBLIC

NoticesandRemarks CopyrightandDistribution
Thisreportisconsideredbyhasheyetobepublicinformation;itislicensedtoaprivate clientunderthetermsoftheprojectstatementofworkandhasbeenmadepublicatthe client'srequest.Materialwithinthisreportmaynotbereproducedordistributedinpartor inwholewithouttheexpresswrittenpermissionofhasheye. 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 2Prysmv3.2.2SecurityAssessment PUBLIC

TableofContents

ExecutiveSummary EngagementOverview
AprivateclientengagedhasheyetoreviewthesecurityofPrysm.FromMarch13toApril
7,2023,ateamoftwoconsultantsconductedasecurityreviewoftheclient-provided
sourcecode,witheightperson-weeksofeffort.Detailsoftheproject’stimeline,testtargets,
andcoverageareprovidedinsubsequentsectionsofthisreport. ProjectScope
Ourtestingeffortswerefocusedontheidentificationofflawsthatcouldresultina
compromiseofconfidentiality,integrity,oravailabilityofthetargetsystem.Weconducted
thisauditwithfullknowledgeofthesystem.Wehadaccesstothefullpublicsourcecode
anddocumentation.Weperformedbothautomatedandmanualprocessesofthetarget systemanditscodebase.
SummaryofFindings Theauditdidnotuncoveranysignificantflawsordefects.Onlyonenotablefindingwas
found;however,itismediumseverityandrepresentsanissuewiththeEthereum
Specification,notwiththePrysmsoftwareitself. EXPOSUREANALYSIS SeverityCount Medium2 Low4
Informational4 Undetermined5 CATEGORYBREAKDOWN CategoryCount AccessControls1 AuditingandLogging1
Configuration1 DataExposure1 DataValidation7 DenialofService1 Documentation1 hasheye
5Prysmv3.2.2SecurityAssessment PUBLIC

Timing1 UndefinedBehavior1 NotableFindings
Notableflawsthatimpactsystemconfidentiality,integrity,oravailabilityarelistedbelow. ● TOB-PRYSM-8
Denial-of-service(DoS)attackscanbelaunchedagainstablockproposerinorderto
forcetheblockproposertomissitsproposalslot.Thisfindingisnotexclusiveto
Prysm,anditaffectsallEthereumconsensusclientimplementationsuptoand includingtheBellatrixhardfork.
hasheye 6Prysmv3.2.2SecurityAssessment PUBLIC

ProjectSummary ContactInformation Thefollowingmanagerswereassociatedwiththisproject:
DanGuido,AccountManagerBrookeLanghorne,ProjectManager dan@hasheye.iobrooke.langhorne@hasheye.io
Thefollowingengineerswereassociatedwiththisproject:
BenjaminSamuels,ConsultantDominikCzarnota,Consultant
benjamin.samuels@hasheye.iodominik.czarnota@hasheye.io DamilolaEdwards,Consultant
damilola.edwards@hasheye.io ProjectTimeline
Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent March6,2023Pre-
projectkickoffcall March17,2023Statusupdatemeeting#1 March24,2023Statusupdatemeeting#2
March31,2023Statusupdatemeeting#3 April10,2023Deliveryofreportdrafttoclient&Prysmteam
April10,2023Reportreadoutmeeting May3,2023Deliveryoffinalreport
October23,2023Deliveryoffinalreportwithfixreview hasheye 7Prysmv3.2.2SecurityAssessment PUBLIC

ProjectGoals Theengagementwasscopedtoprovideasecurityassessmentof Prysm.Specifically,we
soughttoanswerthefollowingnon-exhaustivelistofquestions: ●
DoesPrysmcorrectlyimplementprotectionstopreventvalidatorsfrombeing slashed? ●
CanDoSattacksbeusedtodisruptaPrysmvalidator’soperation? ●
DoesPrysm’simplementationadheretotheEthereum2.0specification? ●
DoesPrysmproperlyvalidateallinputstoprotectagainstpotentialissues? ●
WhatistheoptimalsystemconfigurationtorunaPrysmvalidatorsafelyand securely? ●
Whatkindsofcontrolsandproceduresshouldalarge-scalevalidatoroperator
followtorunPrysmvalidatorssafelyandsecurely? ●
HoweffectivearePrysm’sfuzzingharnesses,andwherecantheybeimproved? ●
IsPrysmvulnerabletoRogueBLSkeyattacks? hasheye 8Prysmv3.2.2SecurityAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthefollowingtarget: Prysm
Repositoryhttps://github.com/prysmaticlabs/prysm Version
v3.2.2(e2fa7d40e3f496416283cc1d329a8ff6c048cb8a) TypeBlockchainConsensusSoftware
PlatformCrossplatform hasheye 9Prysmv3.2.2SecurityAssessment PUBLIC

ProjectCoverage Thissectionprovidesanoverviewoftheanalysiscoverageofthereview,asdeterminedby ourhigh-levelengagementgoals.Ourapproachesincludethefollowing: • Validatoravailability.Webeganbyidentifyingpotentialentrypointsfor DoS attacksandcross-referencingthoseentrypointswiththeconsensusspecification. ManualanalysiswasperformedagainstthespecificationtoidentifyDoSissuesthat maypreventavalidatorfromsubmittingattestationsorproposingblocks. Next,weperformedlightthreatmodelingtounderstandthecrypto-economic incentivesforperformingaDoSattackagainstavalidator. Finally,wereviewedtheconfigurationandthrottlingcapabilitiesforgossip endpointstoassesswhethermaliciouspeerswouldbepenalized/dropped. • Slashingprotection.WeconductedamanualreviewofPrysm’ssafeguardsfor preventingthevalidatorfrombeingslashed.Thesesafeguardswere cross-referencedwiththeconsensusspecification’srecommendedbestpracticesfor avoidingslashing. • Specificationadherence.Weconductedamanualreviewtocomparethe consistencyanddifferencesbetweentheEthereum2.0consensusspecificationand thePrysmcodebase.Wetrackedsectionsofthecodebasethatwereimplemented inaccordancewiththeofficialspecifications,takingintoaccountthechangesacross forksandvariationsbetweenprogramminglanguagesusedinthespecification pseudocodeandthematchingimplementationinGolang.Weobservedthatthe codebaseappearstohavecorrectlyimplementedtheprotocol'sblockchainstate transition,slot,epochprocessing,andattestationrules.Wheredifferencesexisted, theywereprimarilyduetochangesinfunctionnaming,inclusionofadditional checks,errorhandling,andtheuseofhelperfunctionswithintheimplementation code. • Implementationquality.Prysm’scodequalitywasreviewedusinganarrayofstatic tooling,detailedinappendixH.Aftertriagingthefindingsraisedbythetooling,we conductedamanualreviewagainsthotspotsinthecodebasethatthetooling flaggedaspotentiallyindicativeofagreaterproblem. • Cryptography.AbriefreviewofPrysm’suseofcryptographywasperformedby reviewingitsrandomnesssources,encryptionconfiguration,andkeymanagement code.TheEthereumConsensusspecificationwasreviewedtoseeifitisvulnerable tocommonBLSattackssuchasroguekeyattacks. hasheye 10Prysmv3.2.2SecurityAssessment PUBLIC

• Fuzzinganalysis.AreviewofPrysm’sfuzzingharnesseswasconductedto investigatetheircodecoverageandeffectiveness.Wedevelopedscriptstorunthe fuzzingharnessesandgeneratecodecoveragereportsfromthem.Wealsoadded threenewfuzzingharnessesandextendedthecodecoverageofexistingharnesses thatdealtwithgossip/pubsubtopics.Weranthefuzzingharnessesforarounda weekandreportedtheissuesfoundwiththem.AppendixGdescribesthesetupof theautomateddynamicanalysistoolsandtestharnessesusedduringthisaudit. CoverageLimitations Becauseofthetime-boxednatureoftestingwork,itiscommontoencountercoverage limitations.Thefollowinglistoutlinesthecoveragelimitationsoftheengagementand indicatessystemelementsthatmaywarrantfurtherreview: • CompatibilitywithotherEthereumProofofStakeconsensusclients • Correctnessofthecryptographicprimitivesusedduringnodeoperationand encryptionatrest(BLS,ECDSA,BIP39,ERC-2335,etc.).Theseoperationsare implementedbythird-partylibrariesoutofthescopeoftheaudit. • CorrectnessandimplementationoftheBeaconChainValidatorAPI • Depositcontractswerenotaudited. • TheaccuracyofPrysm’simplementationoftheEthereum2.0consensus specificationwasverifiedwiththebesteffortthatcouldbeachievedduringa time-boxedengagement.Wemanuallyreviewedonlythecorecomponentsofthe beacon-chainimplementation,includingthecoderelatedtoblockprocessing,state transitions,validatoroperations,decodingofgossipmessages,andassociated helperfunctions. • OutsidethecontextofDoSissues,theEthereum2.0Consensusspecificationitself wasnotauditedforsecurity/correctness. • IntegrationswithblockconstructionservicesandtheirsecurityimplicationsfOr validatoroperatorswerenotreviewed. • Prysm’sthird-partydependencies,theirmaintenancestatus,andtheirtimesince lastupdatewerenotreviewed. hasheye 11Prysmv3.2.2SecurityAssessment PUBLIC

AutomatedTesting hasheyeusesautomatedtechniquestoextensivelytestthesecuritypropertiesof software.Weusebothopen-sourcestaticanalysisandfuzzingutilities,alongwithtools developedinhouse,toperformautomatedtestingofsourcecodeandcompiledsoftware. TestHarnessConfiguration Weusedthefollowingtoolsintheautomatedtestingphaseofthisproject: ToolDescriptionPolicy gotest-fuzzGo’sbuilt-incoverageguidedfuzzingengine.AppendixG SemgrepAnopen-sourcestaticanalysistoolforfindingbugsand enforcingcodestandardswheneditingorcommitting codeandduringbuildtime AppendixH CodeQLAcodeanalysisenginedevelopedbyGitHubto

automatesecuritychecks AppendixH Go-secAstaticanalysisutilitythatlooksforavarietyof problemsinGocodebases.Notably,go-secwillidentify potentialstoredcredentials,unhandlederrors, cryptographicallytroublingpackages,andsimilar problems. AppendixH Go-vetApopularstaticanalysisutilitythatsearchesformore go-specificproblemswithinacodebase,suchas mistakespertainingtoclosures,marshaling,andunsafe pointers.Go-vetisintegratedwithinthegocommand itself,withsupportforothertoolsthroughthevettool commandlineflag. AppendixH StaticcheckAstaticanalysisutilitythatidentifiesbothstylistic problemsandimplementationproblemswithinaGo codebase.Notethatmanyofthestylisticproblems staticcheckidentifiesarealsoindicativeofpotential AppendixH hasheye 12Prysmv3.2.2SecurityAssessment PUBLIC

“problemareas”inaproject. IneffassignAstaticanalysisutilitythatidentifiesineffectual assignments.Theseineffectualassignmentsoften identifysituationsinwhicherrorsgounchecked,which couldleadtoundefinedbehavioroftheprogramdueto executioninaninvalidprogramstate. AppendixH ErrcheckAstaticanalysisutilitythatidentifiessituationsinwhich errorsarenothandledappropriately. AppendixH gokartAstaticanalysistoolforGothatfindsvulnerabilities usingthesinglestaticassignment(SSA)formoftheGo sourcecode. AppendixH goleakAGoroutineleakdetectorthatdetectsleaksbydirectly hookingintoaproject’steststuite. AppendixH GCheckAsuiteofstaticdetectorstoanalyzeGosoftware,for exampletofindgoroutineleaks. Weattemptedto useitbutitfailed asitdoesn't supportnewGo modulesprojects. AreasofFocus Ourautomatedtestingandverificationworkfocusedondetectingthefollowingissues: • Generalcodequalityissuesandunidiomaticcodepatterns • Issuesrelatedtoincorrecterrorhandlingorincorrectdatavalidation • PotentialgoroutineleaksthatcouldleadtoDoSscenarios hasheye 13Prysmv3.2.2SecurityAssessment PUBLIC

CodebaseMaturityEvaluation hasheyeusesatraffic-lightprotocoltoprovideeachclientwithaclearunderstandingof theareasinwhichitscodebaseismature,immature,orunderdeveloped.Deficiencies identifiedhereoftenstemfromrootcauseswithinthesoftwaredevelopmentlifecyclethat shouldbeaddressedthroughstandardizationmeasures(e.g.,theuseofcommonlibraries, functions,orframeworks)ortrainingandawarenessprograms. CategorySummaryResult ArithmeticPrysmroutesmanymathematicaloperationsthrougha dedicatedmodule, math_helper.go .Thismodule checksforinvalidoperations,overflows,underflows,and divide-by-zeroconditions. However,morescrutinyshouldbeappliedtoensurethat allimportantmathoperationsactuallyusethemath helper,orelsecodeshouldbedocumentedtojustifywhy itdoesnotneedtousethemathhelper(forpurposes suchasthecomputationofattestationdeltas). Satisfactory AuditingPrysmprovidesauditing/monitoringcapabilitiesfor monitoringtheactivevalidatororothervalidatorsonthe network.Loggingcapabilitiescoverallcriticalvalidator events,andimportanteventsarespecificallycalledoutin Prysm’sdocumentation. PrysmalsoexposeslocallyaccessiblePrometheus metricsalongwithsomeGrafanadashboardsfor monitoringoneormorevalidators. Strong Authentication/ AccessControls Thedistributedaspectofthesystemmeansthatcertain traffic,realizedbythegossippeer-to-peerprotocol,does notrequireauthorization. Thebeaconchainallowsauthenticationwiththe executionendpointanditsbeaconchainAPIendpoint. However,wefoundanendpointthat,whenexplicitly enabled,doesnotrequireauthenticationwhileitshould, Strong hasheye 14Prysmv3.2.2SecurityAssessment PUBLIC

orelseitshouldberemovedorredesigned,asdetailedin TOB-PRYSM-9. Complexity Management Prysmtakesgoodmeasurestomanagethecomplexityof theconsensusclient.Splittingtheconsensusclientinto thebeaconchainandvalidatorclientsmadethecode mucheasiertoreview. Thecodealsocontainsinlinereferencestotheoriginal consensusspecification,whichmakeiteasierto understandandreasonabouthowPrysmworks. Strong ConfigurationPrysmprovidesextensivesecureconfigurationoptions forendusers.However,insomeplacesthe documentationdoesprovidebestpracticesforusers (TOB-PRYSM-14). WhilePrysmdoesprovideoptionsforstrong authenticationatitsvariousendpoints,insomecases thereissupportforallowingsecretvaluestobepassed toPrysmoverCLI,creatingariskofdisclosure (TOB-PRYSM-3). Moderate Cryptography andKey Management Prysm’suseofcryptographyandkeymanagementaligns withcurrentbestpractices.Allmeaningfulsourcesof entropycomefromcryptographicallysecurerandom numbergenerators. TheunderlyingimplementationforBLS12-381uses blst , awell-knownandauditedlibrary.Prysmcodewrapping blst functionsiswelldocumentedandincludes informationaboutassumptionsanddesign considerations. PrysmsupportsstoringkeymaterialremotelyinanHSM viaweb3signer. Strong DataHandlingPrysmtakesnecessaryprecautionswhenvalidatingmost typesofincomingdata.Manypartsofthecodehave cleardocumentationthatcallsoutwhenapieceofdata isvalidatedandwhatkindsofvalidationsoccurinagiven function(e.g., ConvertToIndexed , VerifyBlockSignatureUsingCurrentFork). Moderate hasheye 15Prysmv3.2.2SecurityAssessment PUBLIC

However,somepartsofthecodebasehavemuchless robustdatahandling,asweobservedinTOB-PRYSM-11, TOB-PRYSM-12,andTOB-PRYSM-15. DocumentationPrysmprovidesacomprehensivesetofsetup, configuration,backup,operations,andmonitoring documentation.Inparticular,Prysm's"SecurityBest Practices"documentationishighqualityandoffersa comprehensiveguideonhowtorunavalidatorsecurely. Strong MaintenancePrysmhasthoroughinlinecodedocumentationthat comprehensivelyoutlinesspecialconsiderationsand assumptionsforcriticalcodepaths.Prysm'scomments alsoincludeinlinecodesnippetsfromtheconsensus specification,whichgreatlysimplifythecomparisonof specificationandimplementation. However,Prysm'sunittestsarechallengingto successfullyrun,assomeofthemdependonavery specificnetworkconfiguration.Thismaymakelong-term maintenancemorechallengingasmaintainersrotatein andoutoftheproject. Prysmlackshigher-leveldeveloperdocumentationthat describesthesystemasawhole(e.g.,howthevalidator clientandbeaconclientinteractwitheachotherduring attestation),makingauditingoftheprojectmore challenging. Satisfactory MemorySafety andError Handling PrysmiswrittenexclusivelyinGo,soisnotexposedto memorysafetyissues. Prysmhandleserrorsproperlyinthevastmajorityofits codebase.Theonlyexceptionisseveralbenignissuesin unittests(TOB-PRYSM-1). Strong Testingand Verification Prysmhasthoroughunittestcoverageknown-goodtest cases;however,itlacksunittestscoveringvariouserror conditionsthesoftwaremayencounter. Althoughend-to-endtestsarepresent,theycanberun onlyonaPrysm-managedbuildserver. Satisfactory hasheye 16Prysmv3.2.2SecurityAssessment PUBLIC

PrysmusesseveralstaticanalysistoolsaspartofitsCIto catchsimplebugsandlow-qualitycode(e.g.,errcheck, unused,gocognit,unused) Prysmhasanarrayoffuzzingharnessesthatwere reviewedduringtheaudit.Manyarerunaspartofthe unittestsuite,butatleastoneharnessappearstobe unmaintainedandfailed(TOB-PRYSM-13). hasheye 17Prysmv3.2.2SecurityAssessment PUBLIC

SummaryofRecommendations hasheyerecommendsthatPrysmaticLabsaddressthefindingsdetailedinthisreport andtakethefollowingadditionalstepstoenhanceitssecurityposture: • Negativeunitteststotestcriticalcodecomponentsthatmustfailundercertain conditions(e.g.,incorrectsignatureverification,attestationusingthewrongbitmap slot) • AddCodeQLrulestodetectwhenimportantabstractionsarebypassed(e.g., someonecreatesafilewithoutusing fileutil). • Codethatdeviatesfromthereferencespecificationshouldbedifferentiallyfuzzed againstacanonicalimplementation(seefigureG.12inAppendixG:Automated DynamicAnalysis). • Addadditionalhigh-leveldeveloperdocumentationtohelponboardnew contributors.Examplesofsuchdocumentationinclude: ○ Additional README fileswithdeveloperdocumentationineachmajormodule directory(e.g., prysm/validator , prysm/beacon-chain , prysm/tools) ○ Systemdiagramswithreferencestolocationswherespecific implementationsmaybefound ○ Sequencediagramsofcriticalcodepathsandhowtheytraverse Prysm's variouscomponents ○ Annotationsforfunctionsthatmayonlybeusedbytests,whichwillprevent thesefunctionsfrombeingconfusedwithoraccidentallyusedinproduction implementations(ex: funcRandKey() in crypto/bls/bls.go) ○ Standardizedterminologyforfunctionnames(e.g.,verifyvs.validatevs. IsValid) • Worktowardsstandardizingunit/e2etestharnessestoenabledeveloperstorunall testsfromtheirlocalmachines.Thisprocessmaybesimplifiedbyrunningtests insideastandardizedcontainerenvironment. hasheye 18Prysmv3.2.2SecurityAssessment PUBLIC

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails. IDTitleTypeSeverity 1UnhandledErrorsAuditingand Logging Informational 2os.Create()usedwithoutcheckingforanexisting file DataValidationInformational 3Passingsensitiveconfigurationvaluesthroughthe commandlinemayleaktootherprocessesonthe system DataExposureLow 4Theconfigurationfilesarenotcheckedfor permissionswhiletheymaycontainsensitive values AccessControlsLow 5Thebeacon-chainandvalidatorRPCAPIscan panicwhichisrecoveredbutmayleadtocrashes duetomemoryexhaustion DataValidationLow 6GoroutineleaksUndefined Behavior Undetermined 7PotentialdeadlockiftheFeed.Sendpanicis recoveredandthefunctionisretried TimingUndetermined 8BlockProposerDDoSDenialofServiceMedium 9Thedbbackupendpointmaybetriggeredvia SSRForwhenvisitinganattackerwebsitewhich maycauseadenialofservice DataValidationMedium 10MaximumgRPCmessagesizeofMaxInt32(2GB) setinbeacon-chain/servermayleadtodenialof service ConfigurationInformational hasheye 19Prysmv3.2.2SecurityAssessment PUBLIC

11EpochParticipation.UnmarshalJSONmayparse invaliddata DataValidationUndetermined 12Uint256.UnmarshalJSONmayparseinvaliddataDataValidationUndetermined 13FailedassertionsintheFuzzExecutionPayload fuzzingharness DataValidationUndetermined 14TheJWTauthenticationdocsshouldnotsuggest generatingthesecretwiththeuseofthird-party websites DocumentationLow 15PotentiallyinsufficientgossiptopicvalidationDataValidationInformational hasheye 20Prysmv3.2.2SecurityAssessment PUBLIC

DetailedFindings 1.Unhandlederrors Severity:InformationalDifficulty:N/A Type:AuditingandLoggingFindingID:TOB-PRYSM-1 Target: various Description

ThePrysmtestscontainmultipleineffectualassignmentsto err errorvariables.These errorsareneverchecked,whichmayleadtoundefinedtestbehaviorgivenasufficiently largechangeset. Ineffectualassignmentsto err canbeviewedinfigures1.1-1.5.
wsb,err:=blocks.NewSignedBeaconBlock(knownBlocks[i])
err=WriteBlockChunk(stream,chain,p2.Encoding(),wsb)
iferr≠nil&&err.Error()≠network.ErrReset.Error(){ require.NoError(t,err) } Figure1.1:Theerrorfrom blocks.NewSignedBeaconBlock isoverwritten. (prysm/beacon-chain/sync/rpc_send_request_test.go#242-246) bc:=params.BeaconConfig() altairSlot,err:=slots.EpochStart(bc.AltairForkEpoch)
bellaSlot,err:=slots.EpochStart(bc.BellatrixForkEpoch) require.NoError(t,err)
Figure1.2:Theerrorfrom slots.EpochStart online136isoverwritten. (prysm/encoding/ssz/detect/configfork_test.go#135-138)
bellav:=bytesutil.ToBytes4(params.BeaconConfig().BellatrixForkVersion)
altairS,err:=slots.EpochStart(params.BeaconConfig().AltairForkEpoch)
bellaS,err:=slots.EpochStart(params.BeaconConfig().BellatrixForkEpoch) require.NoError(t,err)
Figure1.3:Theerrorfrom slots.EpochStart online207isoverwritten. (prysm/encoding/ssz/detect/configfork_test.go#206-209)
bellav:=bytesutil.ToBytes4(params.BeaconConfig().BellatrixForkVersion) hasheye

altairS,err:=slots.EpochStart(params.BeaconConfig().AltairForkEpoch)
bellaS,err:=slots.EpochStart(params.BeaconConfig().BellatrixForkEpoch) require.NoError(t,err)
Figure1.4:Theerrorfrom slots.EpochStart online298isoverwritten. (prysm/encoding/ssz/detect/configfork_test.go#297-300)
signedTx,err:=types.SignTx(tx,types.NewLondonSigner(chainid),key) iferr≠nil{ returnnil }
err=backend.SendTransaction(context.Background(),signedTx) returnnil Figure1.5:Theerrorfrom backend.SendTransaction isoverwritten. (prysm/testing/endtoend/components/eth1/transactions.go#115-120) ExploitScenario AlargesetofchangesismadetoPrysmthatcausesoneoftheoverwrittenerrorsto becomematerial,inadditiontointroducingabugthattheaffectedtestshoulddetect. Sincetheerrorgoesuncaught,thetestmaycompletesuccessfullydespitethesoftwarenot performingtheintendedbehavior. Recommendations Shortterm,addcodetocatchtheunhandlederrors. Longterm,addautomatedanalysistoolingtoPrysm’sCIpipelinesuchas ineffassign . hasheye

2.os.Create()usedwithoutcheckingforanexistingfile Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-PRYSM-2 Target: prysm/tools/interop/convert-keys/main.go prysm/tools/specs-checker/download.go Description Prysmuses os.Create() inseverallocationswithoutcheckingforthepresenceofan existingfile.Ifafileisalreadypresentwhen os.Create() iscalled,theoriginalfilewillbe truncated.Theaffectedcodesectionsuse os.Create() withtheexpectationthatan existingfilewillcauseanerror,orthatanyexistingfilewillbeoverwritten.
outFile,err:=os.Create(os.Args[2]) iferr≠nil{
log.WithError(err).Fatalf("Failedtocreatefileat%s",os.Args[2]) } Figure2.1: os.Create usedtocreateafilewithoutcheckingforitspresencefirst (prysm/tools/interop/convert-keys/main.go#55-58) funcgetAndSaveFile(specDocUrl,outFilePathstring)error{ //Createoutputfile.
f,err:=os.Create(filepath.Clean(outFilePath)) iferr≠nil{
returnfmt.Errorf("cannotcreateoutputfile:%w",err) } Figure2.2: os.Create usedtocreateafilewithoutcheckingforitspresencefirst (prysm/tools/specs-checker/download.go#41-46) ExploitScenario Thespeccheckerisrunataspecificpointintime,thenagainlaterafterthespecchanges. Thespecfilesarecreatedusing os.Create() ,whichtruncatestheoriginalfileandcreates aninvalidspec. Recommendations Shortterm,replacecallsto os.Create() with os.OpenFile("file.txt", os.O_CREATE|os.O_EXCL,0600) .Thisalternatecallwillgenerateanerrorifthefile alreadyexists. hasheye

Longterm,considerabstractingcallsto os.Create() throughPrysm’sdedicatedfile managementmodulein prysm/io/file/fileutil.go .Inaddition,addachecklistfor pullrequestreviewsforthereviewertoexplicitlycheckfornew os.Create() callssothat guidancemaybegiventousethe fileutil.go module. hasheye 24Prysmv3.2.2SecurityAssessment PUBLIC

3.Passingsensitiveconfigurationvaluesthroughthecommandlinemayleak tootherprocessesonthesystem Severity:LowDifficulty:High Type:DataExposureFindingID:TOB-PRYSM-3 Target:Commandlineflags Description The --execution-endpoint anditsprevious(anddeprecated) --http-web3provider Prysmconsensusnode(beacon-chain)commandlineflagsallowuserstospecifyan authenticationheader(figure3.1).Specifyingsensitivevaluesthroughcommandlineflags mayleaktheirvaluestoanattackerwhocanreadfilesonthemachine.Thiscanhappen when: • Theattackercanlistsystemprocesseseitherwiththe psaux commandorby readingrelevantfilesinthe /proc/ filesystem. • Anattackerfindsanarbitraryfilereadvulnerabilityinanotherapplicationrunning

onthesystemandreadsthe /proc/$pid/cmdline flagofPrysmprocesses. • Theprocesscommandlineflagsareexposedtoamonitoringserviceandthe attackergetsaccesstoitsdata. --execution-endpointvalueAnexecutionclienthttp endpoint.Cancontainauthheaderaswellintheformat(default: "http://localhost:8551") --http-web3providervalueDEPRECATED:Amainchain web3providerstringhttpendpoint.Cancontainauthheaderaswellintheformat --http-web3provider="https://goerli.infura.io/v3/xxxx,Basicxxx"forprojectsecret (base64encoded)and--http-web3provider="https://goerli.infura.io/v3/xxxx,Bearer xxx"forjwtuse(default:"http://localhost:8551")
Figure3.1:Thecommandlineflagsthatmaybeusedtosetanauthheader
Inadditiontothosetwocommandlineflags,thereareothersthatmaybeusedtopass sensitivevalues,forexample: --execution-headers or --grpc-headers .
Alsopleasenotethatthevisibilityoftheprocesscommandlineflagsmaybereducedby usingthehidepid=2andgid=0mountoptionsfortheprocfilesystem.However,this optionisnotenabledbydefaultonthemajority(orall?)ofLinuxdistributionsandisusually notusedbysystemadministrators. hasheye 25Prysmv3.2.2SecurityAssessment PUBLIC

ExploitScenario Anattackerfindsanarbitraryfilereadvulnerabilityinawebapplicationthatrunsonthe sameserverasaPrysmnode.Theyexploitthevulnerabilitytoreadthecommandline argumentsfromthePrysmnodeandfindtheconfiguredexecutionendpointURLalong withitsauthenticationheader,sincetheuserpasseditthroughthecommandlineflag.The attackerusesthisinformationtofurtherattacktheexecutionclient. Recommendations Shortterm,removetheoptiontopasstheexecutionendpointauthenticationheader throughacommandlineflagandinstructtheuserstoprovideanysensitiveconfiguration values,or,evenalltheconfigurationthroughaYAMLconfigfilesetwiththe --config-file flag. Longterm,regularlyauditthecommandlineoptionsofallPrysmbinariestoensurethey disallowordiscourageusersfromprovidingsensitivevalues.Also,documentthe risksof passingsensitivedatathroughthecommandlineflags. hasheye 26Prysmv3.2.2SecurityAssessment PUBLIC

4.Configurationfilescontainingpotentiallysensitivevaluesarenotchecked forpermissions Severity:LowDifficulty:High Type:AccessControlsFindingID:TOB-PRYSM-4 Target:configurationfilesloading Description ThePrysmbinariesdonotcheckthattheconfigurationfilesarenotreadablebyother usersonthesystem,butthesefilesmaycontainsensitivevalues,suchastheexecution endpointauthenticationheadervalue(figure4.1).Thismayallowanattackertoreadand usethesevaluesiftheconfigurationfilesaresetwithoverlybroadpermissions. $ls-la total16 drwxr-xr-x4dcstaff128Mar1712:28. drwxr-xr-x74dcstaff2368Mar1605:09.. -rwxrwxrwx1dcstaff17Mar1712:28config.yaml -rwxrwxrwx1dcwheel66Mar1605:02jwt.hex $bazelrun//cmd/beacon-chain:beacon-chain----config-file=`pwd`/config.yaml --jwt-secret=`pwd`/jwt.hex ... INFO:Buildcompletedsuccessfully,1totalaction [2023-03-1712:30:53]INFOFinishedreadingJWTsecretfrom /Users/dc/src/github.com/prysmaticlabs/prysm/chain/jwt.hex ... [2023-03-1712:30:53]INFOnode:Startingbeaconnodeversion=Prysm/Unknown/Local build.Builtat:Momentsago [2023-03-1712:30:53]INFOblockchain:BlockchaindataalreadyexistsinDB, initializing... [2023-03-1712:30:53]INFOgateway:StartingAPImiddleware ...
Figure4.1:Theblockchainstartsandoperateswithoutensuringthattheconfigurationfiles cannotbereadbyotherusers.
AlthoughthePrysmnodedoescheckfilepermissionsforthewalletpasswordfileandwhen itwritesfiles,thisprotectiondoesnotextendtoverifyingthepermissionsofits configurationfileswhentheyareread.. hasheye 27Prysmv3.2.2SecurityAssessment PUBLIC

ExploitScenario AnattackercanreadfileslocatedontheserveronwhichaPrysmnoderunsasadifferent userthantheonewhorunsthenode.TheattackerreadsthePrysmnodeconfigurationfile sinceitwassettobeworld-readable.Theattackerthenusesthesecretvaluesreadfrom thisfile,suchastheexecutionendpointauthenticationheader,tofurtherattacktheuser's Prysmnode.Notethattheattackerdoesnotnecessarilyrequireshellaccesstoreadfiles, sincetheycouldfindanothervulnerabilitythatallowsthemtodoso. Recommendations Shortterm,checkthepermissionsoffilesanddirectoriesthatmaycontainsensitivedata andwarntheuserandexittheprogramiftheyareoverlybroad.Thewarningshould suggestthattheuserrotateanysecretsthatthefilesmayhavestored,sincethesefiles mayhavebeenreadbyotherusersonthesystem.Suchacheckshouldbeimplementedby usingthe File.Stat functiononanalreadyopenedfile,sinceperformingthecheck beforeopeningthefileleadstotime-of-checkvstime-of-use(TOCTOU)issues.
Longterm,addteststoensurethePrysmbinarieschecktheconfigurationfileshave overlybroadpermissions. hasheye 28Prysmv3.2.2SecurityAssessment PUBLIC

5.Panicsbythebeacon-chainandvalidatorRPCAPIscanpanicarerecovered butmayleadtocrashesduetomemoryexhaustion Severity:LowDifficulty:High Type:DataValidationFindingID:TOB-PRYSM-5 Target: prysm/api/pagination/pagination.go,APIendpoints Description Thebeacon-chainRPCAPIsthatsupportpaginationviathe pagination.StartAndPage functionpanicduetoanout-of-boundssliceindexingwhentheyreceiveanegativevalueof the PageToken or PageSize parameters.SuchpanicsarelaterrecoveredbythegRPC serverandsodonotcrashthenode.However,thisissuemaystillallowanattackerto causeadenialofservice(DoS)ofthenode(e.g.,throughmemoryexhaustionwhenmany invalidrequestsaresent). Figures5.1-2showexamplerequeststhatareinterceptedandmodifiedusingtheBurp Suiteproxy.Intheseexamples,avalidatorrunwiththe --rpc flagpanics,causingtheAPI toreturna500InternalServerErrorresponsecontainingthepanicerror. Figure5.1:Sendingarequestwith pageToken=-1 causesapanicanda500response. hasheye 29Prysmv3.2.2SecurityAssessment PUBLIC

Figure5.2:Sendingarequestwith pageSize=-5 causesapanicanda500response. Figure5.3showswhereoneoftheAPIendpointspanics,butalltheotherendpoints implementsimilarlogic.The ListAttestations endpointcalls StartAndEndPage and passesinthe req.PageToken and int(req.PageSize) .Ifthesevaluesaresentas negativenumbers,theywillcausethe StartAndEndPage functiontoreturnanegative start value(figure5.4),whichisthenusedbythe ListAttestations functiontoindex the atts slice.
```
func(bs*Server)ListAttestations(/*(...)*/){ //(...)
start,end,nextPageToken,err:=pagination.StartAndEndPage(req.PageToken,
int(req.PageSize),numAttestations) //(...) return&ethpb.ListAttestationsResponse{
Attestations:atts[start:end],//⟵out-of-boundsslicepanic TotalSize:int32(numAttestations),
NextPageToken:nextPageToken, },nil } Figure5.3: prysm/beacon-
chain/rpc/prysm/v1alpha1/beacon/attestations.go#L100-L105
funcStartAndEndPage(pageTokenstring,pageSize,totalSizeint)(/*(...)*/){ //(...)
token,err:=strconv.Atoi(pageToken) iferr≠nil{/*(...)*/} //Startpagecannotbegreaterthansetsize.
```
hasheye 30Prysmv3.2.2SecurityAssessment PUBLIC

```
start:=token*pageSize//⟵tokenorpageSizemaybenegative ifstart⩾totalSize{/*(...)*/}
//Endpagecannotgooutofbound. end:=start+pageSize//⟵bothtokenandpageSizemaybenegative //(...)
returnstart,end,nextPageToken,nil } Figure5.4: prysm/api/pagination/pagination.go#L22-L42
```
TheAPIendpointsthatarevulnerabletothisissueareasfollows: • Inbeacon-chain: ListValidatorAssignments , ListAttestations , ListIndexedAttestations , AttestationPool , ListBeaconBlocks (through listBlocksForEpoch and listBlocksForSlot), ListValidatorBalances , ListValidators • Invalidator: ListAccounts Recommendations
Shortterm,fixtheAPIendpointspaginationlogictorejectnegativevaluesbyimplementing thefollowingfixes: 1.Changetherequests PageSize fieldstypesfrom int32 to uint32 . 2.Changethe StartAndEndPage functiontoparsethe pageToken stringasa32-bit unsignedintegerthroughthe strconv.ParseUint function. ThiswillpreventtheAPIendpointsfrompanicsthatcouldpotentiallyleadtocrashes.
Longterm,addteststoensurethatthepaginationAPIrejectsnegativevalues. hasheye 31Prysmv3.2.2SecurityAssessment PUBLIC

6.GoroutineleakscanleadtoDenialofService Severity:UndeterminedDifficulty:High Type:UndefinedBehaviorFindingID:TOB-PRYSM-6 Target: tests Description ManytestsinthePrysmcodebaseleakgoroutines,meaningthatthegoroutinesspawned bytestsdonotfinishandhangonanIOwait,channelreceiveorsend,oraselectstate.As aresult,theresourcesusedbythosegoroutinesarenotfreed.Ifthesegoroutineleakscan happencontinuouslythroughoutthenormalPrysmnoderun,or,iftheycanbetriggered byanattacker,thiscanleadtoDoSthroughresourceexhaustion. Wedetectedthisissuebyusingtheuber-go/goleakpackageandaddingormodifyingthe TestMain functioninallPrysmpackagesthatincludedteststocallthe goleak.VerifyTestMain(m) function.Weincludethelistofallgoroutinesthatleakedas partofoneormanyofthetestsinappendixE. Theseverityofthisfindingisundetermined,aswehavenotanalyzedorconfirmedwhether anyofthosegoroutinescanleakcontinuouslyinanormalrunofaPrysmnodeorwhether suchaleakcanbetriggeredbyanattacker. Recommendation Shortterm,addtheuber-go/goleakgoroutineleaksdetectiontoPrysmtestsand investigatethegoroutineleaksthatoccur.Fixallgoroutineleaksthatarevalid,and documentandignore(byspecifyingtheminacallto VerifyTestMain)anythatarefalse positives. hasheye 32Prysmv3.2.2SecurityAssessment PUBLIC

7.PotentialdeadlockiftheFeed.Sendpanicisrecoveredandthefunctionis retried Severity:UndeterminedDifficulty:High Type:TimingFindingID:TOB-PRYSM-7 Target: prysm/async/event/feed.go Description The Feed.Send functioncancauseadeadlockifitiscalledagainonthesameobjectafter thefirstcallpanicsandisrecovered. Theseverityofthisfindingisundetermined,aswehavenotconfirmedifthiscanhappenin practice.However,thefunctionshouldunlockthe f.mu mutexbeforepanickingtoprevent

thepotentialdeadlock. //Senddeliverstoallsubscribedchannelssimultaneously. //Itreturnsthenumberofsubscribersthatthevaluewassentto. func(f*Feed)Send(valueinterface{}) (nsentint){ //(...) //Addnewcasesfromtheinboxaftertakingthesendlock. f.mu.Lock() f.sendCases=append(f.sendCases,f.inbox...) f.inbox=nil if!f.typecheck(rvalue.Type()){ f.sendLock← struct{}{} panic(feedTypeError{op:"Send",got:rvalue.Type(),want:f.etype}) } f.mu.Unlock() Figure7.1:prysm/async/event/feed.go#L141-L149 Recommendations Shortterm,fixthelackof f.mu unlockinthe Feed.Send functionbeforepanic.Thiswill preventthedeadlockthatcouldhappenifthe Feed.Send functioniscalledagainaftera panicfromitspreviouscall. hasheye 33Prysmv3.2.2SecurityAssessment PUBLIC

8.BlockProposerDDoS Severity:MediumDifficulty:Medium Type:DenialofServiceFindingID:TOB-PRYSM-8 Target:Ethereum2.0Specification Description TheEthereum2.0Specificationdefinesa“blockproposer”foreachbeaconchainslot[1]. Blockproposersareresponsibleforselectingasetoftransactionstoincludeintheblock thateachsatisfiesthebeaconchain’sstatetransitionfunction.Whenvalidatorssuccessfully proposeavalidblock,theyarerewardedanamountofETHthatconstitutesalarge percentageofvalidatorstakingrewards.Theblockproposersforagivenslotinanepoch canbepre-calculateduptooneepochaheadoftime. TheIPaddressofagivenvalidator(andthus,theIPofablockproposer)canbedetermined byanyP2Pactorusingpreviouslypublisheddeanonymizationattacks[2]. Giventhattheidentityofblockproposersisknownaheadoftime,andthatablock proposer’sIPaddressmaybeascertainedbyamaliciousactor,blockproposersmaybe subjecttodistributeddenial-of-service(DDoS)attackswiththegoalofforcingtheblock proposertomisstheirproposalslot. Thisattackcanbeextendedtopreventvalidatorsfromfulfillingothervalidator responsibilities,suchasslotattestations.Butpreventingblockproposalsismorelikelyasit mayhaveafinancialincentive,asdemonstratedintheexploitscenariobelow. ThisattackisnotuniquetoPrysm,andispresentinallEthereum2.0consensusclientsat thetimeofwriting. ExploitScenario AttackerAliceandvictimBoboperateEthereumvalidators.Intheupcomingepoch,Bobis assignedtoproposeablockinslot1,andAliceisassignedtoproposeablockinslot2. WhilewaitingforBobtoproposeablock,Alicenoticesatransactiononthegossiplayer thatpaysanextraordinarilyhighpriorityfee,andthatthistransactionwilllikelybeincluded inBob’sblock. AlicethenlaunchesaDDoSattackagainstBob’svalidatorinordertoknockitofflinesoit willmissitsproposalslot.Bob’svalidatormissesitsproposalslot,andAliceincludesthe hasheye 34Prysmv3.2.2SecurityAssessment PUBLIC

high-valuetransactioninherproposedblockforslot2.Throughthisattack,Aliceeffectively stealsaproportionofthevalidatorawardsthatweredesignatedforBob’sslot. Recommendation Shortterm,theonlyknownmitigationistoconfigurevalidatorswitha “front-end/”back-end”networktopology[3].Inthistopology,eachvalidatorclient(back end)routesitsblockproposalsandattestationstooneoftwobeaconchainnodes(front end).Usingthismitigation,anattackagainstablockproposerwilltakedownthe attestationbeaconchainnode(asthatistheIPaddressthatcanberevealedusingcurrent research),andtheblockproposalnodeisfreetorelaytheproposedblocktotherestofthe network. Itshouldbenotedthatthismitigationwouldrequirethevalidatorandeachbeaconchain nodetobedeployedtoseparateservers,whichcomplicatesdeploymentandviolatesthe currentrecommendationtorunallthevalidationsoftwareonasinglemachine. Theremaybeothermitigations,suchasusingasacrificialreverseproxyforattestation gossipordeployinganL3firewallinfrontofthevalidator.However,duetotime constraints,wecouldnotverifytheeffectivenessofsuchmitigations. Longterm,theEthereumFoundationshouldpursuesecretleaderelectionsforinclusionin afuturehardfork. References [1]EthereumPhase0Spec-BlockProposal [2]PracticalDeanonymizationAttackinEthereumBasedonP2PNetworkAnalysis [3]HowIlearnedtostopworryingabouttheDoSandlovethechain hasheye 35Prysmv3.2.2SecurityAssessment PUBLIC

9.ThedbbackupendpointmaybetriggeredviaSSRForwhenvisitingan attackerwebsite,whichmaycauseaDoS Severity:MediumDifficulty:High Type:DataValidationFindingID:TOB-PRYSM-9 Target: /db/backup monitoringAPIendpoint Description APrysmnode(beacon-chainorvalidator)runwiththe --enable-db-backup-webhook flagexposesa GET/db/backup monitoringAPIendpointthatsavesabackupofthenode databaseontothedisk.AlthoughthemonitoringAPIishostedonlocalhost(127.0.0.1)by default,anattackercanstillreachthisAPIeitherbyperformingthisrequestwhentheuser whohostsaPrysmnodevisitstheirsite,or,ifthenodeishostedonaserver,througha serversiderequestforgery(SSRF)vulnerability. ThiscanleadtoaDoSeitherbyfillingupthediskspaceorbycausinganout-of-memory scenarioasdetailedinthedatabasebackupsdocumentation(sincethebackuploadsthe

wholedatabaseintomemoryfirst).Thismayalsoleavethenodedatabasefilescorrupted. Additionally,notethatthereisnoauthenticationforthemonitoringAPI;ifitwereexposed tothepublic(bysettingthe --monitoring-host flag)withthedatabasebackupsenabled, itwouldbemucheasiertoexploitthisvulnerability. Forthesakeofcompleteinformation,thedefaultmonitoringAPIendpointportsare8080 and8081forthebeacon-chainandvalidator,respectively. ExploitScenario APrysmnodeishostedwithdbbackupswebhookenabledonaserverthatrunsotherweb applications.Anattackerfindsabuginoneofthewebapplicationsthatallowstheattacker toperformanSSRFattack.Theattackerforgesasmanyrequeststo thePrysmnode monitoringendpointaspossibleinordertocreatedbbackupsandfilluptheserverdisk space,causingthePrysmvalidatortobeslashedoncethenodestopsoperating. Recommendation Shortterm,removethe GET/db/backup endpoint.Thisendpointisnotonly unauthenticatedandpotentiallyreachableusingcross-siterequestforgery(CSRF)orSSRF attacks,butitisalsounsafetouse,asitmaycauseanout-of-memoryscenarioandcorrupt thedatabase. hasheye 36Prysmv3.2.2SecurityAssessment PUBLIC

10.MaximumgRPCmessagesizeofMaxInt32(2GB)setin beacon-chain/servermayleadtoDoS Severity:InformationalDifficulty:High Type:ConfigurationFindingID:TOB-PRYSM-10 Target: beacon-chain/server Description ThePrysm3.2.2releasechangedthemaximumgRPCmessagesizeinthe beacon-chain/serverprogramtoavalueof2GB(figure10.1).Thismayallowanattacker whocansendrequeststothisservertocauseaDoSbysendingverylargemessages. Figure10.1:Thepatchthatchangedthemaximumgrpcmessagesize (github.com/prysmaticlabs/prysm/pull/12072) Toclarify,thebeacon-chain/serverisnotthebeacon-chainclientnode.The beacon-chain/serverisaproxyserveronecanusetoexposethebeacon-chain'sgRPCAPI throughanHTTP/JSONRESTAPI.Effectively,withthedefaultconfiguration,the beacon-chain/serverlistenson 127.0.0.1:8000 andcanforwarditsrequeststo localhost:4000 ,whichisthebeacon-chaingRPCAPI. Recommendation Shortterm,changethedefaultmaximumgRPCmessagesizetoasanesmallvalue. Alternatively,warntheuserinthelogswhentheyrunthebeacon-chain/serverona non-localhostaddress.Thiswillhelpuserstoavoidthisissuewhen,forexample,they decidetohostthebeacon-chain/serveronapublicaddressbutdonotreadthewarning fromthe grpc-max-msg-size flagdescription. Longterm,alwayssetsafedefaultsandrequireuserstochangethemifneeded. hasheye 37Prysmv3.2.2SecurityAssessment PUBLIC

11.EpochParticipation.UnmarshalJSONmayparseinvaliddata Severity:UndeterminedDifficulty:High Type:DataValidationFindingID:TOB-PRYSM-11 Target: prysm/beacon-chain/rpc/apimiddleware/structs_marshalling.go Description The EpochParticipation.UnmarshalJSON functionusedtodeserializeaJSONbyteslice doesnotvalidatetheformatoftheprovidedJSONstring,assumingthatitstartsandends withquotationmarks,whichmaynotbetrue.Forexample,unmarshallingavalueof []byte("XdHJ1ZQ==X") withunexpected X charactersatthebeginningandendingwould notreturnanerror,althoughanerrorwouldbeexpected. ThismayleadtoasituationwhereamalformedJSONstringissuccessfullyparsedasthe epochparticipationvaluewhenitshouldberejectedinstead.Thiscanfurthercauseother problems(e.g.,iftheJSONstringwouldbeparsedbytwodifferentsystems). Theseverityofthisfindingisundeterminedbecausewehavenotexhaustivelyanalyzed thiscase.
```
//EpochParticipationrepresentsparticipationofvalidatorsintheirduties.
typeEpochParticipation[]string func(p*EpochParticipation)UnmarshalJSON(b[]byte)error{
ifstring(b)=="null"{ returnnil } iflen(b)<2{
returnerrors.New("epochparticipationlengthmustbeatleast2") }
//Removeleadingandtrailingquotationmarks.
decoded,err:=base64.StdEncoding.DecodeString(string(b[1:len(b)-1])) iferr≠nil{
returnerrors.Wrapf(err,"couldnotdecodeepochparticipationbase64value") }
*p=make([]string,len(decoded)) fori,participation:=rangedecoded{ (*p)
[i]=strconv.FormatUint(uint64(participation),10) } returnnil
```
hasheye 38Prysmv3.2.2SecurityAssessment PUBLIC

} Figure11.1: prysm/beacon-chain/rpc/apimiddleware/structs_marshalling.go#L10-L32 Recommendation Shortterm,fixthe EpochParticipation.UnmarshalJSON functionsothatitrejects invalidinputsthatdonotfittheexpecteddataformat.Addadditionaltestcasestotestthe functionbehavior. hasheye 39Prysmv3.2.2SecurityAssessment PUBLIC

12.Uint256.UnmarshalJSONmayparseinvaliddata Severity:UndeterminedDifficulty:High Type:DataValidationFindingID:TOB-PRYSM-12 Target: prysm/api/client/builder/types.go Description The Uint256.UnmarshalJSON function(figure12.1)doesnotvalidatetheformatofthe providedJSONbyteslice.Asaresult,thefunctionhandlescasesthatstartorendwith quotes,whichallowsmalformedJSONvalues,suchasthosebelow,tobeparsed successfully: • "123 • 123"

Theseverityofthisfindingisundeterminedbecausewehavenotexhaustivelyanalyzed thiscase.However,aswithTOB-PRYSM-11,parsingofmalformeddatamayleadto problemsdependingonvariousfactors,includinghowthefunctionisused,whetheritis possibletocontrolthedataittakes,oriftwosystemsthatwouldbefedwiththesamedata wouldbeexpectedtoprocessitinthesameway. func(s*Uint256)UnmarshalJSON(t[]byte)error{ start:=0 end:=len(t) ift[0]=='"'{ start+=1 } ift[end-1]=='"'{ end-=1 } returns.UnmarshalText(t[start:end]) } Figure12.1: prysm/api/client/builder/types.go#L129-L139 Recommendation Shortterm,fixthe Uint256.UnmarshalJSON functionsothatitrejectsinvalidinputsthat donotfittotheexpecteddataformat.Addadditionaltestcasestotestthefunction behavior. hasheye 40Prysmv3.2.2SecurityAssessment PUBLIC

13.FailedassertionsintheFuzzExecutionPayloadfuzzingharness Severity:UndeterminedDifficulty:High Type:DataValidationFindingID:TOB-PRYSM-13 Target: prysm/beacon-chain/execution/engine_client_fuzz_test.go Description Wefoundtwocaseswhenthe FuzzExecutionPayload fuzzingharnessfailsoneofits assertions.ThisfuzzingharnessensuresthattheGeth's engine.ExecutableData structurelayoutmatchesthePrysm's pb.ExecutionPayload structure.Itdoessoby unmarshallingtheinputbytesliceintothetwostructures,comparingtheirerrors,andthen performingadditionalmarshalingoperations. ThefirstinputthatfailsanassertionaboutmismatchedGethandPrysmunmarshallerrors hasa withdrawals fieldthatisnotsupportedbyPrysm'sstructure(figure13.1).Thisfield, addedintheCapellafork,hasbeenaddedtotheGeth'sgo-ethereumpackageinthispatch (see gen_ed.go modifications),anda1.11.0versionincludingitwasreleasedonFebruary 25,2023. ThisissuedoesnotseemtoaffectthesecurityofthePrysmcodebase,sinceitdispatches muchofitslogicbasedontheforkversion,andits ExecutionPayloadCapella structure supportstheCapellaforkandits withdrawals field. f.Add([]byte(" {\"parentHash\":\"0xff01ff01ff01ff01ff01ff01ff01ff01ff0100000000000000000000000000000000\","+ "\"feeRecipient\":\"0xff01ff01ff01ff01ffff01ff01ff01ff01ff0000\","+ "\"stateRoot\":\"0xff01ff01ff01ff01ff01ff01ff01ff01ff0100000000000000000000000000000000\","+ "\"receiptsRoot\":\"0xff01ff01ff01ff01ff01ff01ff01ff01ff0100000000000000000000000000000000\","+ "\"logsBloom\":\"0x6a756e6b"+strings.Repeat("0",504)+"\","+ "\"prevRandao\":\"0xff01ff01ff01ff01ff01ff01ff01ff01ff0100000000000000000000000000000000\","+ "\"blockNumber\":\"0xffffffffffffffff\","+ "\"gasLimit\":\"0xffffffffffffffff\","+ "\"gasUsed\":\"0xffffffffffffffff\","+ "\"timestamp\":\"0x64\","+ "\"extraData\":\"0x\","+ "\"baseFeePerGas\":\"0x7fffffffffffffff\","+ "\"blockHash\":\"0xff01ff01ff01ff01ff01ff01ff01ff01ff0100000000000000000000000000000000\","+ "\"transactions\":["+ "\"0xff01ff01ff01ff01ff01ff01ff01ff01\","+ "\"0xff01ff01ff01ff01ff01ff01ff01ff01\","+ "\"0xff01ff01ff01ff01ff01ff01ff01ff01\","+ "\"0xff01ff01ff01ff01ff01ff01ff01ff01\"],"+ "\"withdrawals\":9999}")) Figure13.1:Firstinputthattriggersanassertioninthe FuzzExecutionPayload harness hasheye 41Prysmv3.2.2SecurityAssessment PUBLIC

Thesecondinputreturnsan"emptyhexstring"erroronthePrysmunmarshallerbutno errorontheGethunmarshaller.Thisisduetotheempty baseFeePerGas field,asshown infigure13.2. f.Add([]byte(" {\"pArentHAsh\":\"0X0000000000000000000000000000000000000000000000000000000000000000\","+ "\"feeReCipient\":\"0X00000000000000000000000000000000000000\","+ "\"stAteRoot\":\"0X0000000000000000000000000000000000000000000000000000000000000000\","+ "\"reCeiptsRoot\":\"0X0000000000000000000000000000000000000000000000000000000000000000\","+ "\"logsBloom\":\"\","+ "\"prevRAndAo\":\"0X0000000000000000000000000000000000000000000000000000000000000000\","+ "\"BloCkNumBer\":\"\","+ "\"gAsLimit\":\"\","+ "\"gAsUsed\":\"\","+ "\"timestAmp\":\"\","+ "\"eXtrADAtA\":\"\","+ "\"BAseFeePerGAs\":\"\","+ "\"BloCkHAsh\":\"0X0000000000000000000000000000000000000000000000000000000000000000\","+ "\"trAnsACtions\":[]}")) Figure13.1:Thesecondinputthattriggersanassertioninthe FuzzExecutionPayload harness Recommendations Shortterm,doublecheckthatthedescribedissuecannotaffectproductionPrysmcode. Also,fixthecodeorfuzzingharnesssoitdoesnotcrashonthedescribedinputsand assertion. Longterm,runthefuzzingharnessesregularlyandfixanybugstheyfind,especiallybefore Prysmreleases. hasheye 42Prysmv3.2.2SecurityAssessment PUBLIC

14.TheJWTauthenticationdocssuggestgeneratingthesecretusing third-partywebsites Severity:LowDifficulty:High Type:DocumentationFindingID:TOB-PRYSM-14 Target:https://docs.prylabs.network/docs/execution-node/authentication Description ThePrysm'sJWTauthenticationdocumentationsuggeststheJWTtoken'ssecretvaluecan begeneratedusingthird-partywebsites: Useanonlinegeneratorlikethis.Copyandpastethisvalueintoa jwt.hex file. Thismayleadtoasituationwherethesecretgeneratedbytheuserisleakedorused againsttheirnodebecauseathird-partyserviceusedbytheuserstoredthesecretsor generatedanon-

randomvalue. ExploitScenario
Thesuggestedonlinegeneratorwashackedandnowservespredictablerandomvalues.
TheattackerfindsaGethinstancethatusesauthenticationandusesthepredictable
randomvaluetoauthenticate.Theythenusetheauthenticatedendpointtocauseadosof thenode. Recommendations
Shortterm,donotrecommendtheuseofthird-partywebsitestogenerateauthentication
secretvaluesintheJWTauthenticationdocumentation,sincesuchwebsitesmayproduce non-
randomvaluesorstorethegeneratedvalues,addingariskthatthesecretmayleakor beusedagainsttheuser.
AnalternativesolutionistohostsuchatokensecretgeneratorwithinPrysmitself,which
eliminatestheneedforuserstotrustathird-partyservicetogeneratethetoken. hasheye
43Prysmv3.2.2SecurityAssessment PUBLIC

15.Potentiallyinsucientgossiptopicvalidation Severity:InformationalDifficulty:High
Type:DataValidationFindingID:TOB-PRYSM-15 Target:Multiplecodepaths Description
SomecodepathsinthePrysmprojectprocesspubsub/gossipmessagestopicsina
potentiallyinsufficientmanner.Whileothervalidationsinotherprocessingstepsappearto
preventthesecodepathsfromintroducingsecurityrisks,additionalcodechangesorthe
introductionofnewgossipmessageformatscouldmakethosecodepathsproblematic.
Theissueisthataccordingtothep2pinterfaceconsensusspecs,thepubsubtopicsmust
conformtothefollowingformat: /eth2/ForkDigestValue/Name/Encoding
ThePrysmprojectdealswiththesepubsubtopicsinthefollowingways: 1.Inthe decodePubsubMessage
function,itdoesnotvalidateorcheckthatthe Encoding partis ssz_snappy
.Instead,itonlytrimsthisencodingfromthetopic
andleavesthetopicstringintactiftheencodingpartisnotthere.Thisisrealizedby thefollowingcodeline:
topic=strings.TrimSuffix(topic,s.cfg.p2p.Encoding().ProtocolSuffix())
2.Multiplefunctionscheckforagivenpubsubtopicusingthe strings.Contains(topic,<somestringliteral>)
code,whichcanleadto incorrecttopicattributionifatopicofagivenkindcouldcontainastringusedby
anotherkindoftopic.Thefollowingfunctionsdemonstratethisbehavior: • decodePubsubMessage •
topicScoreParams • addDigestToTopic • addDigestAndIndexToTopic • updateMetrics • registerRPCHandler
WehavediscussedthesepotentialissueswiththePrysmteamtovalidatewhetherthey
mayaffectthefunctioningofPrysmnodes.APrysmnodewouldnotacceptamessagewith hasheye
44Prysmv3.2.2SecurityAssessment PUBLIC

anunexpectedpubsubmessagetopic,sincethemessageswillbefilteredbytheirtopicsby
thislibp2plibrary,accordingtothefiltersetbythebeacon-chaincode. Recommendation
Shortterm,considerrefactoringthecodeofthe decodePubsubMessage andother
functionsthatdealwithpubsubmessagetopicstocentralizeandsimplifythevalidationof
pubsubtopics.Avoidchoosingthemessagetopicwiththeconstructionslike strings.Contains(topic,
<somestringliteral>) . Longterm,implementend-to-endtestsorfuzzingharnessesthatwillvalidatethatall
pubsubtopicsthatdonotfitintotheexpectedformatsareproperlyrejectedbythe system. hasheye
45Prysmv3.2.2SecurityAssessment 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
46Prysmv3.2.2SecurityAssessment 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 47Prysmv3.2.2SecurityAssessment PUBLIC

B.CodeMaturityCategories
ThefollowingtablesdescribethecodematuritycategoriesandratingcriteriausedinthisdocumentCodeMaturityCategories 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 48Prysmv3.2.2SecurityAssessment PUBLIC

WeakManyissuesthataffectsystemsafetywerefound.
MissingArequiredcomponentismissing,significantlyaffectingsystemsafety.
NotApplicableThecategoryisnotapplicabletothisreview.
NotConsideredThecategorywasnotconsideredinthisreview. Further Investigation Required
Furtherinvestigationisrequiredtoreachameaningfulconclusionhasheye 49Prysmv3.2.2SecurityAssessment PUBLIC

C.SystemDiagram FigureC.1:SystemdiagramofaPrysmvalidator hasheye 50Prysmv3.2.2SecurityAssessment PUBLIC

D.SecurityGuidanceforOperators
OperatinganEthereumvalidatorcomeswithtechnicalandsecurityrisks;ifthenodeisnot adequatelyoperatedandsecured,thevalidatormaybeslashedandlosefunds.This appendixprovidesguidanceonhowtomitigatethosetechnicalandsecurityrisks.
ThisguidancebuildsonthePrysmteam'ssecuritybestpracticesdocumentationandis focusedonanenterprisesettingwhereasingleactoroperatesmultiplevalidators(referred toasaswarm).
ClientSelection Ethereumproof-of-stakeencouragesclientdiversityinordertoproducethemostrobust networkpossible.Clientdiversityisincentivizedusingthenetwork’s“correlationpenalty” calculationsand“inactivityleak”penalty.
However,inpractice,operatorsmustconsidervariousrisksandfactorswhenchoosingthe validatorclientsoftwaretouse,includingthefollowing: ● Thedifferencebetweenpotentialpenaltiesforrunningabuggymajorityoraless frequentlyusedclient.SeeCorrelationPenaltyandInactivityLeakbelowformore details. ● Thepotentialmitigationsrelatedtosocialconsensus;forexample,abugina majorityclientthatcausesalargenumberofstakerstobeslashedmayleadtoa largerdiscussionand/oractionsbytheecosystem. ● Theclientteam'sreputation ● Thekindsofchangemanagementpracticescurrentlyusedtovalidatenewcode ● TherisksofcompromisefortheclientsteamCorrelationPenalty
Whenavalidatorisslashed,threepenaltiesarelevieda gainstthevalidator’sstake.These includeaninitialpenaltythatreducestheoffender’sstakeby1/32,attestationinactivity penaltiesthatareappliedu ntiltheoffender’swithdrawalepoch,andacorrelationpenalty.
Thecorrelationpenaltyisusedtoapplyextrapunishmentbasedonhowmanyother slashingeventsoccurredafter theoffenderwasinitiallyslashed.Theideabehindthe correlationpenaltyistoscaleslashingpunishmentssothatcoordinatedattacksaremuch moreheavilypenalizedthanone-offevents. hasheye 51Prysmv3.2.2SecurityAssessment PUBLIC

Thecorrelationpenaltycreatesanexistentialthreatforvalidatorsusingaclientthatisused byalargefractionofthevalidatornetwork.Ifabuginavalidatorclientcausestheclientto

commitslashingoffenses,thenthemorevalidatorsthatusethatclient,themoresevere theslashingwillbe.Twotheoreticalsituationsandtheirconsequencesareoutlinedbelow.
1.Aclientisusedby33%ofthevalidatornetwork.Abugintheclientcauseseach validatorrunningittocommitaslashableoffense.Overthefollowingfewhours, everyvalidatorusingtheproblematicclientisslashed.Duetothehighcorrelationof slashingoffenses,everyslashedvalidatorusingtheclientwillhavetheirentire stakeslashed.
2.Aclientisusedby2%ofthevalidatornetwork.Abugintheclientcauseseach validatorusingtheclienttocommitaslashableoffense.Inthissituation,the correlationpenaltywouldworkouttoabout6%ofthevalidator'soriginalstake. InactivityLeak
Inactivityleakprotocolkicksinwhenthechainfailstofinalizeforfourepochs(overone thirdofthevalidatorsetisoffline)andpenalizesvalidatorsthatareofflinebyslashingtheir stake. Thismeansifaspecificvalidatorclientisusedbyover33%ofthenetworkandencounters abugthatpreventsblocksfrombeingprocessed,allvalidatorsusingthatclientwillbe penalizedbytheinactivityleak. Inactivityleakcreatesadangerforvalidatoroperatorswhoareusingbeacon chain/executionclientsthatareusedbyalargefractionofthevalidatorset.Forthisreason, validatoroperatorsareincentivizedtopreferclientsthatareusedbyaminorityofthe network. CurrentclientdiversitystatisticscanbefoundinfigureD.1. hasheye 52Prysmv3.2.2SecurityAssessment PUBLIC

FigureD.1:ClientdiversitystatisticsasofMarch30,2023 (https://clientdiversity.org/#distribution)
NodeConfiguration DeploymentConfiguration
Thevalidatorclient,beaconchainclient,andexecutionclientshouldbedeployedtoa singlemachineorKubernetespodunlessthereisagoodoperationalreasontodo otherwise.Ifdeploymentacrossmultiplemachinesorpodsisrequired,connections betweeneachclientmustbesecuredusingTLSforencryptionandJWTsforauthentication. Thedeploymentprocessitselfshouldalsobetested,documented,andreproducibleso thatitispossibletoredeploythenodeintheeventofanemergency.
Wealsorecommendkeepingoneormoreadditional"staging"or"test"deployments with thesameexactorsimilarsetup(butwithdifferentkeys,ofcourse)againstatestnetwork. Thoseadditionaldeploymentsshouldbeusedfortestingupdatesandotherscenarios,like theprocessofnodemigrationtoanothermachine.
Inaddition,validatoroperatorsshouldavoidpassinganysensitivevaluesincommandline flagswhenconfiguringPrysm.WhenpassedasCLIflags,sensitivevaluescanbeleakedto otherusersonthesystem(TOB-PRYSM-3)andmaybeingestedbylogging/monitoring software.SensitivevaluesrelatedtoauthenticationshouldbepassedtoPrysmusing configurationfiles,andoptionallystoredinaremotekeymanagementsolution/templated intoin-memoryconfigurationfilesatruntime. hasheye 53Prysmv3.2.2SecurityAssessment PUBLIC

NodeUpdates Theoperatorsshouldtrackthenodesoftwareupdatesthroughtheavailableofficial channels.Thenewreleasechangelogshouldbeinspectedcarefullytounderstandallofits consequencesandanystepsthatneedtobeperformedinordertoapplytheupdate.The updateinstallationshouldbetestedfirstonastaging/testsetup,and,ifeverythingworks asexpected,theninstalledonthemainnetsetup. KeyManagement Ethereumproof-of-stakevalidatorsusetwokeys:avalidationkeyandawithdrawalkey. ValidationKey
Thevalidationkeyshouldbemanagedbyaremoteauthenticated Web3Signerinstance, whichinturnshouldstorethekeyatrestinanHSM,cloudkeymanagementsolution,or softwarekeystoragesolutionsuchasHashiCorpVault.
Giventhataspecifickeymaybeneededbymultiplevalidatornodesduringamigrationor backuprecovery,redundantoperationalprotectionsmustbeimplementedtopreventtwo validatorinstancesfromattemptingtousethesamekeyinvalidationactivitiesatthesame time.Theuseofavalidatorkeybytwodistinctvalidatorclientswillleadtothekeybeing slashed. WithdrawalKey
Thewithdrawalkeyisneededonlytowithdrawavalidator'sstake,andassuch,shouldnot bestoredonvalidatornodes.Withdrawalkeysshouldusethresholdsignaturestoforce withdrawalstorequiretheconsensusofmultipleindependentparties.
PrivatethresholdkeysshouldbestoredindedicatedHardwareSecurityModules(HSM) thatareinturnkeptinseparate,securegeographicallocationsandaccessibleonlyby independent,authorizedparties.
Ifprivatethresholdkeysarestoredoncryptocurrencyhardwarewallets,operatorsshould followthe10RulesfortheSecureUseofCryptocurrencyHardwareWallets.
SlashingProtectionDatabaseManagement
Thebeaconchainnodehasseveralpersistentdatabases,themostimportantofwhichis theslashingprotectionhistory.Thisdatabaseiscriticalforpreventingavalidatorfrom beingslashedandmustbemigratedifavalidatorisre-syncedordeployedtoanother client.
Ifaslashingprotectiondatabaseiscorruptedandcannotberecoveredfrombackup, operatorsmustwaitatleasttwofinalizedepochsbeforerestartingtheirvalidator.Itmust

beemphasizedthatundernocircumstancesshouldtwovalidatornodesusethesamekey atthesametime.Itfollowsthatcomplexfailoverscenariosshouldnotbeattemptedby validatoroperatorswithoutathoroughunderstandingofthehighlyskewedrisk/reward hasheye 54Prysmv3.2.2SecurityAssessment PUBLIC

tradeoffofsuchasystem.Atthistime,therearenoknownvalidatoroperatorssuccessfully usinganysuchfailoversystem,noristheresoftwaresupportforsuchasysteminany publiclyavailableconsensusclient. SlashingNodes Largevalidatoroperatorsshouldavoidrunningtheirvalidatorswithslashingenabled,as thisrequiressignificantlymorediskspace.Instead,operatorsshouldrunoneormore dedicatedslashernodesondifferentmachineswithstaticpeeringarrangementswiththe validatingswarm. Slashernodesarenotrequiredforoperatingavalidatororearningfeesfromslashing penalties.However,bynotrunningaslashingnode,theoperatormusttrustthatanother entityonthenetworkisrunningacorrectlyconfiguredslashernodeandthattheirslashing reportsarepropagatedtotheswarm’sblockproposersinatimelymanner.Failingto includeslashingsinablockproposalmayleadtolostearningsthatwouldhavecomefrom theinclusionofslashing. MonitoringandAlerting Validatornodesmusthavesufficientmonitoringandalertingcapabilitiestoescalate potentialincidentsbeforelossesareincurred.PrysmexposesPrometheusmetricsalong withvariouslogsthatshouldbeingestedbyalogging/monitoringsolution.Operators shouldconsideraddingadditionalinstrumentationtomonitorsystemresources,suchas diskutilization,networkutilization,CPUutilization,andRAMutilization. Specificguidanceonthekindsofeventsthatrequiremanualinterventionisdocumented undertheSwarmOperationsection. NetworkConfiguration Validatornodesusethelibp2plibraryforgossip/blocksync/propagation,andassuch,will needtoexposeseveralportstotheopeninternet.OperatorsshouldconsideraddingaDoS preventionsolutiontotheirnetworktopologytopreventtargetedattackslike TOB-PRYSM-8. Remotemanagementmustbeconductedusingadedicatedcontrolplane(accessibleonly fromaprivatenetwork)toavoidexposingremotemanagementportstothepublic. Syncing FreshnodesshouldconfiguretheirconsensusclientstosyncfromarecentWeak Subjectivitycheckpoint.Syncingaconsensusclientfromthegenesisblockshouldbe consideredunsafeduetolong-rangeattacks.WeakSubjectivitycheckpointsmustbe acquiredbyatrustedsource,preferablyfromfully-syncednodesmaintainedbythesame operator. hasheye 55Prysmv3.2.2SecurityAssessment PUBLIC

SwarmOperation IncidentResponse Large-scalevalidatoroperatorsmusthaveadocumentedandwell-understoodincident responseplaninplace.TheIncidentResponseRecommendationsfromBuildingSecure Contractsprovidesagoodfoundationforanenterpriseincidentresponseplan,andany operatorshouldhaveanswersforeachquestionfromthedocument. Allincidentresponseplansshouldbecomplementedwithasetofdocumentedrunbooks onhowtohandledifferentkindsofpotentialincidents.ThePrysmteamprovidesa mitigationworksheetthatoffersagoodstartingpointforanincidentresponserunbook. Themostrelevantriskeventsarereproducedfromthe Prysmdocumentationbelow,along withsomeadditionalriskeventsthatshouldbeconsideredbylarge-scaleoperators. Foreachriskevent,theoperatorshouldgenerateaplanfor: 1.Howeachriskeventistobedetectedandsubsequentlyescalated 2.Aplaybookforthetriage/mitigationofeachriskevent Riskevents: • Datacenter(wherethevalidatornodeishosted)internetconnectiongoesoffline • Validatorhardwareisphysicallydestroyedorstolenfromthedatacenter • Avalidator’sstoragemediaormemoryfails • Avalidator’sOScrashes • Avalidatorrunsoutofsystemresources(disk,RAM,etc.) • Avalidatorexperiencesunusualdisk/CPU/networkactivity • Theconsensus/executionclienthasabug • Avalidatorinstancetransitionsintoanunexpectedstate • Avalidator’svalidationkeysorwithdrawalkeysareexposedtoanattacker • Avalidatorintheswarmisslashed • Avalidatorneedstobemigratedtoanewmachine • Avalidatorisfailingtoattesttoblocksinatimelymanner • Avalidatorismissingblockproposalslots hasheye 56Prysmv3.2.2SecurityAssessment PUBLIC

• AvalidatorissubjecttoaDoSattack • Thevalidatorswarmneedstobemigratedtoanewconsensus/executionclientin responsetoabug • TheHSMstoringthevalidator’skeyisdestroyedorstolen Thelistofriskeventsaboveisnotexhaustive,andeachvalidatoroperatorshouldinternally identifyadditional,organization-specificrisksthatmayjustifyanincidentresponse. Multiplemembersoftheincidentresponseteamshouldbefamiliarwith,andabletorun, eachplaybook.Theincidentresponseteamshouldalsoconsidersimulatingriskeventsto practiceeachplaybook.Someorganizationsuseatechniquecalled ChaosEngineeringto simulateriskeventsinaproductionenvironment;suchatechniquemaybeespecially usefulforthemaintenanceandoperationofahighlyavailablevalidatorswarm. ChangeManagement

Giventhevalueatriskinalargevalidatorswarm,operatorsshouldadheretoastrict changemanagementpolicy.Changestothesystemshouldbeadequatelytestedbefore beingdeployed.Testingshouldbeconductedinarealistic,multi-clientenvironmentsuch astheEthereumpublictestnets.
EachchangetothesystemshouldbeaccompaniedbymonitoringcriteriaandrollBack stepstopreventchangesfromcausingcatastrophicsystemfailuresoroutages. hasheye 57Prysmv3.2.2SecurityAssessment PUBLIC

E.GoroutinesLeakinginPrysmTests
FigureE.1listsallthegoroutinesthatleakedwhenweranPrysmtestswiththegoleak goroutineleaksdetectorasdescribedinfindingTOB-PRYSM-6. github.com/dgraph-io/ristretto. (*Cache).processItems github.com/dgraph-io/ristretto.(*defaultPolicy).processItems github.com/ethereum/go-ethereum/accounts/abi/bind/backends. (*filterBackend).SubscribeNewTxsEvent.func1 github.com/ethereum/go-ethereum/accounts/abi/bind/backends.(*filterBackend).SubscribePendingLogsEvent.func1 github.com/ethereum/go-ethereum/consensus/ethash.(*remoteSealer).loop github.com/ethereum/go-ethereum/core.(*BlockChain).updateFutureBlocks github.com/ethereum/go-ethereum/core. (*txSenderCacher).cache github.com/ethereum/go-ethereum/core/state/snapshot.(*diskLayer).generate github.com/ethereum/go-ethereum/eth/filters.(*EventSystem).eventLoop github.com/ethereum/go-ethereum/metrics.(*meterArbiter).tick github.com/ipfs/go-log/writer.(*MirrorWriter).logRoutine github.com/libp2p/go-flow-metrics.(*sweeper).run github.com/libp2p/go-libp2p-pubsub. (*GossipSubRouter).connector github.com/libp2p/go-libp2p-pubsub.(*GossipSubRouter).heartbeatTimer github.com/libp2p/go-libp2p-pubsub.(*PubSub).handleSendingMessages github.com/libp2p/go-libp2p-pubsub.(*PubSub).processLoop github.com/libp2p/go-libp2p-pubsub.(*Subscription).Next github.com/libp2p/go-libp2p-pubsub.(*backoff).cleanupLoop github.com/libp2p/go-libp2p-pubsub. (*peerScore).background github.com/libp2p/go-libp2p-pubsub.(*validation).validateWorker github.com/libp2p/go-libp2p/p2p/host/autonat.(*AmbientAutoNAT).background github.com/libp2p/go-libp2p/p2p/host/basic.(*BasicHost).background github.com/libp2p/go-libp2p/p2p/host/peerstore/pstoremem.(*memoryAddrBook).background github.com/libp2p/go-libp2p/p2p/host/pstoremanager.(*PeerstoreManager).background github.com/libp2p/go-libp2p/p2p/host/resource-manager.(*resourceManager).background github.com/libp2p/go-libp2p/p2p/net/connmgr.(*BasicConnMgr).background github.com/libp2p/go-libp2p/p2p/net/connmgr. (*decayer).process github.com/libp2p/go-libp2p/p2p/net/swarm.(*DialBackoff).background github.com/libp2p/go-libp2p/p2p/net/upgrader.(*listener).Accept github.com/libp2p/go-libp2p/p2p/protocol/identify.(*ObservedAddrManager).worker github.com/libp2p/go-libp2p/p2p/protocol/identify.(*idService).loop github.com/libp2p/go-libp2p/p2p/transport/quicreuse. (*listener).Accept github.com/libp2p/go-libp2p/p2p/transport/quicreuse.(*reuse).gc github.com/libp2p/go-yamux/v4.(*Session).AcceptStream github.com/libp2p/go-yamux/v4. (*Session).sendLoop github.com/libp2p/go-yamux/v4.(*Session).startMeasureRTT github.com/libp2p/go-yamux/v4.(*Stream).Read github.com/lucas-clemente/quic-go.(*baseServer).accept github.com/lucas-clemente/quic-go.(*baseServer).run github.com/lucas-clemente/quic-go.(*connection).run github.com/lucas-clemente/quic-go.(*incomingStreamsMap[...]).AcceptStream github.com/lucas-clemente/quic-go.(*packetHandlerMap).runCloseQueue github.com/lucas-clemente/quic-go. (*receiveStream).readImpl github.com/lucas-clemente/quic-go.(*sendQueue).Run github.com/patrickmn/go-cache.(*janitor).Run github.com/paulbellamy/ratecounter. (*RateCounter).run.func1 github.com/prysmaticlabs/prysm/v3/async.RunEvery.func1 github.com/prysmaticlabs/prysm/v3/async/event.TestFeed_Send.func6.1 github.com/prysmaticlabs/prysm/v3/beacon-chain/execution.(*PowchainCollector).latestStatsUpdateLoop github.com/prysmaticlabs/prysm/v3/beacon-chain/monitor.(*Service).monitorRoutine github.com/prysmaticlabs/prysm/v3/beacon-chain/node.(*BeaconNode).Start.func1 github.com/prysmaticlabs/prysm/v3/beacon-chain/p2p/peers/scorers.(*Service).loop github.com/prysmaticlabs/prysm/v3/beacon-chain/rpc/eth/events.(*Server).StreamEvents github.com/prysmaticlabs/prysm/v3/beacon-chain/sync.(*Service).registerHandlers github.com/prysmaticlabs/prysm/v3/beacon-chain/sync.(*Service).subscribeDynamicWithSubnets.func1 github.com/prysmaticlabs/prysm/v3/beacon-chain/sync.(*Service).verifierRoutine github.com/prysmaticlabs/prysm/v3/beacon-chain/sync/initial-sync. (*Service).waitForStateInitialization github.com/prysmaticlabs/prysm/v3/beacon-chain/sync/initial-sync.(*blocksFetcher).stop github.com/prysmaticlabs/prysm/v3/container/leaky-bucket. (*Collector).PeriodicPrune.func1 hasheye 58Prysmv3.2.2SecurityAssessment PUBLIC

github.com/prysmaticlabs/prysm/v3/testing/middleware/engine-api-proxy.(*Proxy).Start github.com/prysmaticlabs/prysm/v3/time/slots.(*SlotTicker).Done.func1 github.com/prysmaticlabs/prysm/v3/time/slots.(*SlotTicker).start.func1 github.com/prysmaticlabs/prysm/v3/validator/db/kv.(*Store).batchAttestationWrites github.com/syndtr/goleveldb/leveldb.(*DB).compactionError github.com/syndtr/goleveldb/leveldb.

(*DB).mCompaction github.com/syndtr/goleveldb/leveldb.(*DB).mpoolDrain
github.com/syndtr/goleveldb/leveldb.(*DB).tCompaction github.com/syndtr/goleveldb/leveldb.
(*session).refLoop go.opencensus.io/stats/view.(*worker).start google.golang.org/grpc.
(*addrConn).resetTransport google.golang.org/grpc.(*ccBalancerWrapper).watcher
internal/poll.runtime_pollWait k8s.io/klog.(*loggingT).flushDaemon net/http.
(*persistConn).writeLoop FigureE.1:Listofgoroutinesthatleakedaspartofoneormultipletests hasheye
59Prysmv3.2.2SecurityAssessment PUBLIC

F.Glossary ConsensusClient:Aclientthatsyncs&validatestheEthereumPoSbeaconchain.
Consensusclientsmaybeconfiguredtopartakeinconsensusactivitiesasavalidator.Many
implementationssplittheconsensusclientintotwosmallerclients-abeaconchainclient andavalidatorclient.
BeaconChainClient:Acomponentofaconsensusclientdedicatedtosyncingthebeacon
chain,validatingconsensus,andpersistingthebeaconchain.
ValidatorClient:Acomponentofaconsensusclientdedicatedtoparticipatingin
consensus.Validatorclientsinteractwithbeaconchainclientsusingtheofficial Beacon
ChainAPI.Validatorclientsarenecessaryonlywhenthenodeisconfiguredtoparticipatein
consensusasavalidatornode.
ExecutionClient:Aclientthatsyncsblocks,listensfortransactions,andexecutes
blocks/transactionsintheEVM.
Node:AcompleteEthereumPoSclientthatsyncsthechaintoalocaldatabaseandmaybe
queriedforthechain’sstate.Nodesverifythatthesyncedchainistheoneagreeduponby
consensusandthatthesyncedchain’sexecutionblocksarevalid.Nodesconsistofa
consensusclientandanexecutionclient.
SlasherNode:Anodethatisconfiguredtologallattestationsonthenetworktodetect
slashingviolations.Detectedslashingviolationsaresharedwiththerestofthenetworkvia gossip.
ValidatorNode:Anodethatisparticipatinginconsensus. hasheye 60Prysmv3.2.2SecurityAssessment PUBLIC

G.AutomatedDynamicAnalysis Thisappendixdescribesthesetupoftheautomateddynamicanalysistoolsandtest
harnessesusedduringthisaudit. Thepurposeofautomateddynamicanalysis
Inmostsoftware,unitandintegrationtestsaretypicallytheextenttowhichtestingis
performed.Thistypeoftestingdetectsthepresenceoffunctionality,allowingdevelopersto
ensurethatthegivensystemadherestotheexpectedspecification.However,these
methodsoftestingdonotaccountforotherpotentialbehaviorsthatanimplementation mayexhibit.
Fuzzingandproperty-basedtestingcomplementbothunitandintegrationtestingby
identifyingdeviationsintheexpectedbehaviorofacomponentofasystem.Thesetypesof
testsgeneratetestcasesandprovidethemtothegivencomponentasinput.Theteststhen
runthecomponentsandobservetheirexecutionfordeviationsfromexpectedbehaviors.
Theprimarydifferencebetweenfuzzingandpropertytestingisthemethodofgenerating
inputsandobservingbehavior.Fuzzingtypicallyattemptstoproviderandomorrandomly
mutatedinputsinanattempttoidentifyedgecasesinentirecomponents.Propertytesting
typicallyprovidesinputssequentiallyorrandomlywithinagivenformat,checkingtoensure
aspecificpropertyofthesystemholdsuponeachexecution. Bydevelopingfuzzingandproperty-
basedtestingalongsidethetraditionalsetofunitand
integrationtests,edgecasesandunintendedbehaviorscanbeprunedduringthe
developmentprocess,whichwilllikelyimprovetheoverallsecuritypostureandstabilityof asystem. Tooling
GosupportsfuzzinginitsstandardtoolchainbeginninginGo1.18thatcanbeused throughthe gotest-fuzz
commandandwhichisalsousedbythePrysmproject.
However,thenativeGofuzzingframeworkstilllackssomeusabilityoruserexperience
features.Analternativecouldbehasheyeforkofgo-fuzzthatextendstheoriginal go-
fuzzprojectandforwhichwealsohavesomehelpertools: • go-fuzz-
utils:helperpackagethatprovidesasimpleinterfacetoproducerandom
valuesforvariousdatatypesandcanrecursivelypopulatecomplexstructuresfrom
rawfuzzdata.Itcanalsobeusedwiththenativegofuzzing. StateofPrysmfuzzing
SincePrysmalreadyusedthenativeGofuzzing,wedecidedtousenativeGofuzzingduring
theauditaswell.Prysmimplements18fuzzingharnesses(figureG.1)thatarerunfora
shortperiodoftimeduringtheirCI/CDbuilds.Prysmalsousesthefuzzbuzz.ioserviceto hasheye
61Prysmv3.2.2SecurityAssessment PUBLIC

fuzztheprojectforalongerperiodoftime;however,theresultsorstatisticsofthisfuzzing
arenotpubliclyavailable. container/trieFuzzSparseMerkleTrie_HashTreeRoot
container/trieFuzzSparseMerkleTrie_MerkleProof container/trieFuzzSparseMerkleTrie_Insert
container/trieFuzzSparseMerkleTrie_VerifyMerkleProofWithDepth beacon-
chain/syncFuzzValidateBeaconBlockPubSub_Phase0 beacon-
chain/syncFuzzValidateBeaconBlockPubSub_Altair beacon-
chain/syncFuzzValidateBeaconBlockPubSub_Bellatrix beacon-chain/p2pFuzzMsgID beacon-
chain/executionFuzzForkChoiceResponse beacon-chain/executionFuzzExchangeTransitionConfiguration

beacon-chain/executionFuzzExecutionPayload beacon-chain/executionFuzzExecutionBlock beacon-chain/state/state-nativeFuzzPhase0StateHashTreeRoot beacon-chain/state/state-nativeFuzzAltairStateHashTreeRoot beacon-chain/state/state-nativeFuzzBellatrixStateHashTreeRoot beacon-chain/state/state-nativeFuzzCapellaStateHashTreeRoot beacon-chain/state/fieldtrieFuzzFieldTrie validator/accountsFuzzValidateMnemonic FigureG.1:ExistingfuzzingharnessesinthePrysmproject.Thefirstcolumnisthepackagein whichtheharnessexists,andthesecondcolumnisthefuzzingharnessfunctionname. ScriptstofuzzPrysm Inordertorunthefuzzingharnessesbyourselves,wepreparedthefollowingscripts, whichcanalsobefoundinfiguresG.2–8: • get-fuzzlist.py :aPythonscriptthatsavesallfuzzingharnesspackagepathsand namesintoa fuzzlist file.Itdoesitbyparsingtheoutputofthe gotest-list . commandsexecutedineachpackage • fuzz-all.sh :runsallfuzzingharnessesfromthe fuzzlist fileusingthe fuzz-one.sh script • fuzz-one.sh :ascripttorunasinglefuzzingharness.Itgets <package-path> and <fuzzing-harness-name> arguments.Italsosets2CPUsforeachharness • kill-fuzz.sh :stopsallfuzzers • cov-all.sh :recompilesallharnessesfrom fuzzlist withcoverageprofilingand runallharnessesagainstalltheirinputs • cov-one.sh :gatherscoveragefromasingleharness.Notethatitmovesthe fuzzer-generatedcorpusfilestothefuzzer's testdata/ directorysowecanrunall corpusinputsagainstthefuzzingharness.ThisisaworkaroundfortheGonative fuzzer’slimitationthatallowsittoruntheharnessonlyagainstinputsfromthe hasheye 62Prysmv3.2.2SecurityAssessment PUBLIC

testdata directory(whenprovidinganargument).Thecoverageprofilesaresaved intothe cover/ directory. • gen-html.sh :readsthecoverageprofilesfromthe cover/ directoryandrenders fuzzingcoverageHTMLreportsintothe html/ directory Thesescriptsareshownbelow: importos #os.system("""find.-typed-execbash-c"cd{}&&gotest-list.&&cd-"\;> ../golist.out2>&1""") data=open('../golist.out').read().splitlines() fuzz={} harnesses=[] forlineindata: ifline.startswith('Fuzz'): harnesses.append(line) elifline.startswith('ok'): directory=line.split('prysm/v3/')[1].split()[0] fuzz[directory]=harnesses harnesses=[] fork,vinfuzz.items(): forharnessinv: print(k,harness) FigureG.2:Pythonscripttocreatethe fuzzlist file.Useitasfollows: python3./get-fuzzlist.py>fuzzlist #!/bin/bash IFS=$'\n' forargsin$(cat./fuzzlist) do IFS=$'' stringarray=($args) ./fuzz-one.sh${stringarray[0]}${stringarray[1]} done FigureG.3:The fuzz-all.sh script hasheye 63Prysmv3.2.2SecurityAssessment PUBLIC

#!/bin/bash OUTPATH="$(pwd)/outputs/$2" cd../$1 gotest-v./-fuzz="^$2\$"-run="^$2\$"-parallel=2-cpu=2>$OUTPATH2>&1& FigureG.4:The fuzz-one.sh script #!/bin/bash IFS=$'\n' forargsin$(cat./fuzzlist) do IFS=$'' stringarray=($args) ./cov-one.sh${stringarray[0]}${stringarray[1]} done FigureG.5:The cov-all.sh script #!/bin/bash #$1=dirwithharnesspackage;$2=harness OUTPATH="$(pwd)/outputs/$2" COVERPATH="$(pwd)/cover/$2.cover" cd../$1 FUZZPATH=./testdata/fuzz/$2 mkdir-p$FUZZPATH2>/dev/null mv$FUZZPATH$FUZZPATH-crashers ln-s$(goenvGOCACHE)/fuzz/github.com/prysmaticlabs/prysm/v3/$1/$2$FUZZPATH gotest-v./-run="^$2/"-cover-coverprofile=$COVERPATH rm$FUZZPATH mv$FUZZPATH-crashers$FUZZPATH cd- FigureG.6:Thecov-one.shscript #!/bin/bash killall-v-9go killall-v-9--regexp'.*test' killall-v-9--regexp'state-.*' FigureG.7:The kill-fuzz.sh script hasheye 64Prysmv3.2.2SecurityAssessment PUBLIC

#!/bin/bash foriin$(ls./cover);do gotoolcover-html=./cover/$i-o./html/$i.html done FigureG.8:The gen-html.sh script Ourimprovementsandnewfuzzingharnesses Inordertofuzzefficientlyandincreasethepossibilityofcatchingbugs,werecommend inspectingthecodecoverageofthefuzzersafterrunningthemforalongerperiodoftime andcheckingifitispossibletoimprovetheircoverage.Thiscanbedonebyeither1)adding moreinputstotheinitialcorpussothatmorepathsareknownforthefuzzerupfrontor2) modifyingthecodesothatinefficientpathsaremockedandarenotexecutedduring fuzzing. Duringtheaudit,werantheexistingandnewfuzzingharnessesforafewdaysandthen analyzedtheircodecoveragebyusingthescriptsincludedabove.Onethingwenoticedis thatsomeharnessesdonotcoverthecodepathsthatprocesscertaingossipmessage topics,asshowninfigureG.9.Weimprovedthisbyaddingadditionalinitialtestcasesto thosefuzzingharnesses.ThepatchthatdoesthiscanbeseeninfigureG.10. Apartfromthis,wehavealsoimplementedthreenewfuzzingharnesseswhosecodewe includeinfiguresG.11–G.12: • FuzzDecodePubsubMessage :afuzzingharnesstotestthedecodingofpubsub messages.Notethatitdoesnotcheckthefulltopicformatbutitcouldanditcouldif theusedfunctionswouldvalidatethewholeformat(whichcouldbeaproblemas describedinTOB-PRYSM-15) • FuzzDeserializeRequestBodyIntoContainer :aharnessthatteststhe processingofdatabycertainAPIendpoints • FuzzDifferentialAttestingIndices :harnessthatensuresthatthe attestation.AttestingIndices functionreturnsdataintheexpectedformat Weranthosefuzzingharnessesforafewdaysaswell,buttheydidnotfindanycrashes. hasheye 65Prysmv3.2.2SecurityAssessment PUBLIC

FigureG.9:Codecoveragefromoneofthefuzzingharnesses.Asshownabove,weneverreached codepathsrelatedtocertainpubsubtopics. diff--gita/beacon-chain/p2p/pubsub_fuzz_test.gob/beacon-chain/p2p/pubsub_fuzz_test.go index6d9408114..9c61a5dfe100644 ---a/beacon-chain/p2p/pubsub_fuzz_test.go +++b/beacon-chain/p2p/pubsub_fuzz_test.go @@-11,8+11,10@@import( ) funcFuzzMsgID(f*testing.F){ -validTopic:=fmt.Sprintf(p2p.BlockSubnetTopicFormat, []byte{0xb5,0x30,0x3f,0x2a})+"/"+ encoder.ProtocolSuffixSSZSnappy -f.Add(validTopic) +for_,format:=rangep2p.AllTopics(){ +validTopic:=fmt.Sprintf(format, []byte{0xb5,0x30,0x3f,0x2a})+"/"+ encoder.ProtocolSuffixSSZSnappy +f.Add(validTopic) hasheye 66Prysmv3.2.2SecurityAssessment PUBLIC

+} f.Fuzz(func(t*testing.T,topicstring){ _,err:=p2p.ExtractGossipDigest(topic) diff--gita/beacon-chain/sync/sync_fuzz_test.gob/beacon-chain/sync/sync_fuzz_test.go index00fdbaf72..2496c687c100644 ---a/beacon-chain/sync/sync_fuzz_test.go +++b/beacon-chain/sync/sync_fuzz_test.go @@-83,7+83,19@@funcFuzzValidateBeaconBlockPubSub_Phase0(f*testing.F){ assert.NoError(f,err) topic=r.addDigestToTopic(topic,digest) +//Usecurrent/validpeeridaspid +f.Add(string(p.PeerID()), []byte("junk"),buf.Bytes(),[]byte(topic)) + +//Addoneinputwithinvalidpeerid f.Add("junk", []byte("junk"),buf.Bytes(),[]byte(topic)) + +for_,topicFormats:=rangep2p.AllTopics(){ +digest,err:=r.currentForkDigest() +assert.NoError(f,err) +topic=r.addDigestToTopic(topicFormats,digest) +f.Add("junk",[]byte("junk"),buf.Bytes(), []byte(topic)) +} + f.Fuzz(func(t*testing.T,pidstring,from,data,topic[]byte){ r.cfg.p2p=p2ptest.NewFuzzTestP2P() r.rateLimiter=newRateLimiter(r.cfg.p2p) @@-164,7+176,19@@funcFuzzValidateBeaconBlockPubSub_Altair(f*testing.F){ assert.NoError(f,err) topic=r.addDigestToTopic(topic,digest) +//Usecurrent/validpeeridaspid +f.Add(string(p.PeerID()), []byte("junk"),buf.Bytes(),[]byte(topic)) + +//Addoneinputwithinvalidpeerid f.Add("junk", []byte("junk"),buf.Bytes(),[]byte(topic)) + +for_,topicFormats:=rangep2p.AllTopics(){ +digest,err:=r.currentForkDigest() +assert.NoError(f,err) +topic=r.addDigestToTopic(topicFormats,digest) +f.Add("junk",[]byte("junk"),buf.Bytes(), []byte(topic)) +} + f.Fuzz(func(t*testing.T,pidstring,from,data,topic[]byte){ r.cfg.p2p=p2ptest.NewFuzzTestP2P() r.rateLimiter=newRateLimiter(r.cfg.p2p) @@-245,7+269,19@@funcFuzzValidateBeaconBlockPubSub_Bellatrix(f*testing.F){ assert.NoError(f,err) topic=r.addDigestToTopic(topic,digest) +//Usecurrent/validpeeridaspid +f.Add(string(p.PeerID()), []byte("junk"),buf.Bytes(),[]byte(topic)) + +//Addoneinputwithinvalidpeerid f.Add("junk", []byte("junk"),buf.Bytes(),[]byte(topic)) + +for_,topicFormats:=rangep2p.AllTopics(){ +digest,err:=r.currentForkDigest() +assert.NoError(f,err) +topic=r.addDigestToTopic(topicFormats,digest) +f.Add("junk",[]byte("junk"),buf.Bytes(), []byte(topic)) +} + f.Fuzz(func(t*testing.T,pidstring,from,data,topic[]byte){ r.cfg.p2p=p2ptest.NewFuzzTestP2P() r.rateLimiter=newRateLimiter(r.cfg.p2p) hasheye 67Prysmv3.2.2SecurityAssessment PUBLIC

FigureG.10:Patchthatincreasesthecodecoverageoffuzzingharnessesthatdealwithgossip messagetopicsbyaddingadditionalinputstotheinitialfuzzingcorpus. funcFuzzDeserializeRequestBodyIntoContainer(f*testing.F){ varbodyJsonbytes.Buffer err:=json.NewEncoder(&bodyJson).Encode(defaultRequestContainer()) assert.NoError(f,err) //Addanexampleinput f.Add(bodyJson.Bytes()) f.Fuzz(func(t*testing.T,data[]byte){ varbodyJsonbytes.Buffer bodyJson.Write(data) container:=&testRequestContainer{} err:=DeserializeRequestBodyIntoContainer(&bodyJson,container) iferr==nil{ err2:=ProcessRequestContainerFields(container) iferr2==nil{ req:=http.Request{ Header:http.Header{}, } //Atthispointweexpectnoerrorsinceifwewereableto //deserializethestruct,weshouldalsobeabletoserializeitback err3:=SetRequestBodyToRequestContainer(container,&req) assert.Equal(t,nil,err3) } } }) } FigureG.11:AfuzzingharnessforsomeAPImiddlewarerequestprocessinglogic.Thisharness shouldbeaddedtothe prysm/api/gateway/apimiddleware/process_request_test.go file. funcFuzzDifferentialAttestingIndices(f*testing.F){ f.Add(8,14,8) f.Fuzz(func(t*testing.T,lengthint,bitsToSetint,committeeLengthint){ //Allowonlyupto64bits iflength⩽0||length>64||committeeLength⩽0||committeeLength>64{ return } bitsList:=bitfield.NewBitlist(uint64(length)) //Setrandom(fuzzergenerated)bitindices variuint64 fori=0;i<uint64(length);i++{ hasheye 68Prysmv3.2.2SecurityAssessment PUBLIC

bit:=bitsToSet&(1<<i) bitsList.SetBitAt(i,bit≠0) } //Createcommitteewithsequentialids committee:=make([]primitives.ValidatorIndex,committeeLength) fori=0;i<uint64(committeeLength);i++{ committee[i]=primitives.ValidatorIndex(i) } got,err:=attestation.AttestingIndices(bitsList,committee) //Theonlyallowederrorcaseiswhenthelengthsdonotmatch iferr≠nil{ assert.NotEqual(t,length,committeeLength) }else{ assert.NotNil(t,got) assert.Equal(t,length,committeeLength) //Ensurepropervalueswerereturned,accordingtothebitsList foridx,value:=rangebitsList.BitIndices(){ assert.Equal(t,got[idx],uint64(value)) } } }) }

FigureG.12:Adifferentialfuzzingharnesstovalidatethe attestation.AttestingIndices functionality.Thisharnessshouldbeaddedtothe prysm/proto/prysm/v1alpha1/attestation/attestation_utils_test.go file. hasheye 69Prysmv3.2.2SecurityAssessment PUBLIC

```
funcFuzzDecodePubsubMessage(f*testing.F){
_,err:=signing.ComputeForkDigest(params.BeaconConfig().GenesisForkVersion, make([]byte,32))
require.NoError(f,err) //Addinputswithallknowntopicformats for_,topic:=rangep2p.AllTopics(){
//TODO/FIXME:Probablyformatthetopicsaccordingly f.Add([]byte(""),topic) }
f.Fuzz(func(t*testing.T,data[]byte,topicstring){ msg:=&pubsub.Message{Message:&pb.Message{}}
msg.Message.Topic=&topic msg.Message.Data=data s:=&Service{
cfg:&config{p2p:p2ptesting.NewTestP2P(t),chain: &mock.ChainService{ValidatorsRoot:
[32]byte{},Genesis:time.Now()}}, } got,err:=s.decodePubsubMessage(msg)
//TODO/FIXME:Ideally,thisshouldprobablyassertallknownerrors ifgot≠nil{ assert.NoError(t,err)
}else{ //ifgotisnil,thereshouldbeanerror assert.NotNil(t,err) } }) }
```

FigureG.13:Afuzzingharnesstotestthe decodePubSubMessage function.Thisharness shouldbeaddedtothe prysm/beacon-chain/sync/decode_pubsub_test.go file. Furtherfuzzingideasandguidance ThePrysmfuzzingharnessescanbefurtherimprovedinthefollowingways: 1.Theharnesses,similarlytounittests,shouldassertasmuchstateaspossible,tobe abletocatchregressionbugsincaseoffutureupdates.Theassertionsshouldalso bedesignedsothatthetestorharnesswouldfailifanewfieldwereaddedintoa structureusedintheharness/test.Togivesomeexamples: a.The FuzzMsgId fuzzingharnessshouldensurethatanon-errordigestvalue matchestheexpectedone,oratleastthatithastheexpectedlengthand conformstotheexpectedformat. hasheye 70Prysmv3.2.2SecurityAssessment PUBLIC

b.The FuzzForkChoiceResponse fuzzingharnessshouldperformits assertionsdifferently,sothatifanewfieldisaddedtothego-ethereum's ForkChoiceResponse.PayloadStatus structureitwouldbedetectedby theharnessasbeingnothandledatall(incasethetestedcodewouldset thatfield). c.Testssuchas TestProcessRewardsAndPenaltiesPrecompute should assertallbalancestates,notonlythosethatareexpectedtobechanged. 2.Thestate-changingprotocols,suchasgossipandsync,couldbefuzzedinsucha waythatthefuzzergeneratesaseriesofoperationsandtheinitialstatechain,and thenitrunsthoseprotocoloperationsortransactionsagainstthechain.However,in ordertobeeffective,suchaharnesswilllikelyrequirecodemodificationsor mockingofcertainfunctionalitytomakethecodeefficientandfullydeterministic (e.g.,notrelyingontime,filesondisk,signingandhashing).Itisalsoveryimportant thatsuchaharnessleveragecodecoverageinstrumentationsothatthefuzzer reusesinterestinginputs,whichwouldincreaseitseffectivenessinfindingedge cases. 3.Differentialfuzzingharnessescouldbeaddedtotestthefunctionalitiesdocumented intheconsensusspecificationagainstotherconsensusnodes.This,however,would requirethecreationofseparateprogramsinbothconsensusnodesoftwaresthat wouldtriggeronlythefunctionalitytobetestedandrequirethefunctionalityto haveasimilarinterfaceinbothsolutions.Suchfuzzingharnessescouldensurethat differentnodesgivethesameexactresults(orerrors)forthesameinputs. hasheye 71Prysmv3.2.2SecurityAssessment PUBLIC

H.AutomatedStaticAnalysis

Thisappendixdescribesthesetupoftheautomatedanalysistoolsusedduringthisaudit. Thoughstaticanalysistoolsfrequentlyreportfalsepositives,theydetectcertaincategories ofissues,suchasmemoryleaks,misspecifiedformatstrings,andtheuseofunsafeAPIs, withessentiallyperfectprecision.Werecommendperiodicallyrunningthesestaticanalysis toolsandreviewingtheirfindings. Semgrep ToinstallSemgrep,weused pip byrunning python3-mpipinstallsemgrep .Weused Semgrepversion1.18.0.TorunSemgreponthecodebase,weranthefollowingintheroot directoryoftheproject: semgrep--config"p/hasheye"--sarif--metrics=off--output semgrep.sarif Wealsoranthetoolwiththefollowingrules(configs): • p/ci • p/security-audit • r/go.lang WerecommendintegratingSemgrepintotheproject'sCI/CDpipeline.Integrateatleastthe ruleswithHIGHconfidenceandthosewithMEDIUMconfidenceandHIGHimpact. Werecommendusinghasheye’ssetofSemgreprules(fromtherepositoryorless preferablyfromtheregistry)anddgryskiruleswhichcanbeusedvia p/semgrep-go-correctness . CodeQL WeinstalledCodeQLbyfollowingCodeQL'sinstallationguide.WeusedCodeQLversion 2.12.4,withGoversion1.20.0. AfterinstallingCodeQL,weranthefollowingcommandtocreatetheprojectdatabasefor thePrysmrepository: codeqldatabasecreatecodeql.db--language=go Wethenranthefollowingcommandtoquerythedatabase:

codeqldatabaseanalyzecodeql.db-ocodeql.sarif --format=sarif-latest-j0--tob-go-all hasheye 72Prysmv3.2.2SecurityAssessment PUBLIC

Weusedour tob-go-all querypack,whichincludesselectedqueriesfromthebuilt-in codeql/go-all querypackanddozensofhasheye'sprivatequeries. CodeQLreportedasubstantialnumberoffindingsthatwedeterminedwerenotworth reportingatthecurrentstageoftheproject.However,westronglyrecommendthatall issuesbereviewedandfixedbythePrysmteam. Moreover,werecommendintegratingCodeQLintotheCI/CDpipeline,usingeitherGitHub AdvancedSecurityorcustomintegration.Pleasenotethattheamountoffalsepositives canbereducedbyrunningonlyhighandvery-highprecisionqueries. Otherstaticanalysistools InadditiontoSemgrepandCodeQL,weusedthefollowingtoolsduringtheauditand highlyrecommendintegratingthemintotheCI/CDpipeline. • Go-sec isastaticanalysisutilitythatlooksforavarietyofproblemsinGo codebases.Notably, go-sec willidentifypotentialstoredcredentials,unhandled errors,cryptographicallytroublingpackages,andsimilarproblems. • Go-vet isaverypopularstaticanalysisutilitythatsearchesformoreGo-specific problemswithinacodebase,suchasmistakespertainingtoclosures,marshaling, andunsafepointers. Go-vet isintegratedwithinthe go commanditself,with supportforothertoolsthroughthe vettool commandlineflag. • Staticcheck isastaticanalysisutilitythatidentifiesbothstylisticproblemsand implementationproblemswithinaGocodebase.Notethatmanyofthestylistic problems staticcheck identifiesarealsoindicativeofpotential"problemareas"in aproject. • Ineffassign isastaticanalysisutilitythatidentifiesineffectualassignments.These ineffectualassignmentsoftenidentifysituationsinwhicherrorsgounchecked, whichcouldleadtoundefinedbehavioroftheprogramduetoexecutioninan invalidprogramstate. • Errcheck isastaticanalysisutilitythatidentifiessituationsinwhicherrorsarenot handledappropriately. • gokart isastaticanalysistoolforGothatfindsvulnerabilitiesusingthesinglestatic assignment(SSA)formoftheGosourcecode. • goleak isaGoroutineleakdetectorthatdetectsleaksbydirectlyhookingintoa project'stestsuiteasdescribedinitsdocumentation(byaddingthe goleak.VerifyTestMain(m) lineintoallofthetests' TestMain functions). PleasealsoseeourblogpostonGosecurityassessmenttechniquesforfurtherdiscussion oftheGo-relatedanalysistools. hasheye 73Prysmv3.2.2SecurityAssessment PUBLIC

I.FixReviewResults Whenundertakingafixreview,hasheyereviewsthefixesimplementedforissues identifiedintheoriginalreport.Thisworkinvolvesareviewofspecificareasofthesource codeandsystemconfiguration,notcomprehensiveanalysisofthesystem. FromSeptember18toSeptember22,2023,hasheyereviewedthefixesandmitigations implementedbythePrysmteamfortheissuesidentifiedinthisreport.Wereviewedeach fixtodetermineitseffectivenessinresolvingtheassociatedissue. Insummary,ofthe15issuesdescribedinthisreport,thePrysmteamhasresolvedeight issuesandhasnotresolvedtheremainingsevenissues.Foradditionalinformation,please seetheDetailedFixReviewResultsbelow. IDTitleStatus 1UnhandlederrorsResolved 2os.Create()usedwithoutcheckingforanexistingfileResolved 3Passingsensitiveconfigurationvaluesthroughthecommandlinemay leaktootherprocessesonthesystem Unresolved 4Configurationfilescontainingpotentiallysensitivevaluesarenot checkedforpermissions Unresolved 5Panicsbythebeacon-chainandvalidatorRPCAPIscanpanicare recoveredbutmayleadtocrashesduetomemoryexhaustion Resolved 6GoroutineleakscanleadtoDenialofServiceUnresolved 7PotentialdeadlockiftheFeed.Sendpanicisrecoveredandthe functionisretried Resolved 8BlockProposerDDoSUnresolved hasheye 74Prysmv3.2.2SecurityAssessment PUBLIC

9ThedbbackupendpointmaybetriggeredviaSSRForwhenvisiting anattackerwebsite,whichmaycauseaDoS Unresolved 10MaximumgRPCmessagesizeofMaxInt32(2GB)setin beacon-chain/servermayleadtoDoS Unresolved 11EpochParticipation.UnmarshalJSONmayparseinvaliddataResolved 12Uint256.UnmarshalJSONmayparseinvaliddataResolved 13FailedassertionsintheFuzzExecutionPayloadfuzzingharnessResolved 14TheJWTauthenticationdocssuggestgeneratingthesecretusing third-partywebsites Resolved 15PotentiallyinsufficientgossiptopicvalidationUnresolved hasheye 75Prysmv3.2.2SecurityAssessment PUBLIC

DetailedFixReviewResults TOB-PRYSM-1:Unhandlederrors ResolvedinPR#12578andPR#12938.ThePrysmteamhasaddressedallidentified instancesofunhandlederrorsandmodifiedthecodetoproperlyhandleerrorvalues.Also, theconfigurationfileforthe ineffassign staticcodeanalysistoolhasbeenrefactored, removingthe only_files sectionandretainingonlythe exclude_files section,allowing

ittomoreeffectivelyexcludespecificfilesfromanalysisbasedonregularexpressions. TOB-PRYSM-2:os.Create()usedwithoutcheckingforanexistingfile ResolvedinPR#12536ThePrysmteamhasupdatedthefilecreationlogicwithinthe tools/interop/convert-keys/main.go and tools/specs-checker/download.go files,optingfortheuseof os.OpenFile .Thischoiceincorporatesthe os.O_CREATE and os.O_EXCL flags,whichguaranteethecreationofanewfilewhenitisabsentand safeguardsagainstunintentionaloverwritesofexistingfiles,therebyminimizingpotential datalossorconflictsduringfilecreation. TOB-PRYSM-3:Passingsensitiveconfigurationvaluesthroughthecommandlinemay leaktootherprocessesonthesystem Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Thiswillnotbehandledatthecurrentmomentasanychangestohowconfigfilesare handledandhowsecretsarehandledcanbreakusersintheshortterm.Alsosome hardeningstepscancauseissuesfornon-technicalusersrunningnodes,howeverwewill keepontrackingthissowecanfindanacceptablesolutiontoit. TOB-PRYSM-4:Configurationfilescontainingpotentiallysensitivevaluesarenot checkedforpermissions Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Thiswillnotbehandledatthecurrentmomentasanychangestohowconfigfilesare handledandhowsecretsarehandledcanbreakusersintheshortterm.Alsosome hardeningstepscancauseissuesfornon-technicalusersrunningnodes,howeverwewill keepontrackingthissowecanfindanacceptablesolutiontoit. TOB-PRYSM-5:Panicsbythebeacon-chainandvalidatorRPCAPIscanpanicare recoveredbutmayleadtocrashesduetomemoryexhaustion ResolvedinPR#12932.ThePrysmteamhasupdatedthepagepaginationlogic;input validationchecks wereaddedtoensurethatboth pageSize and totalSize are non-negative,withinformativeerrormessagesincasesofinvalidinput.Additionally,a validationcheckforthe token valuederivedfromthe pageToken wasimplementedto preventnegativevalues. hasheye 76Prysmv3.2.2SecurityAssessment PUBLIC

TOB-PRYSM-6:GoroutineleakscanleadtoDenialofService Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Wehaveelectedtonotfixthisfornowasalotoftheseleaksareduetotest-setupand handlingthemisn'tstraightforward.Howeverwewillthinkofalong-termsolutioninthe futureonhowtohandlethem. TOB-PRYSM-7:PotentialdeadlockiftheFeed.Sendpanicisrecoveredandthe functionisretried ResolvedinPR#12464.ThePrysmteamhasupdatedthe Send functioninthe async/event/feed.go filetoincludeamutexunlockbeforethepanicistriggered. TOB-PRYSM-8:BlockProposerDDoS Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Thereisnowaytofixthisonourend,itrequiresaprotocollevelmitigationratherthan somethingonPrysm'send. TOB-PRYSM-9:ThedbbackupendpointmaybetriggeredviaSSRForwhenvisitingan attackerwebsite,whichmaycauseaDoS Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Thiscan'tbeaddressedimmediatelyasitisabreakingchange,theonlytimewecando thisisduringamajorversionbump(nexthardfork).Sowearewaitingtillthentoenable this. TOB-PRYSM-10:MaximumgRPCmessagesizeofMaxInt32(2GB)setin beacon-chain/servermayleadtoDoS Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Therpcendpointsareassumedtoonlybeexposedtotrusted/peersentities.Forthat reasonwewillnotbechangingthesizeasanyonewithaccesstotherpcendpointscan causeDOSduetothemanythingstheendpointsprovide(stateregeneration,returning fullstatesbacketc). TOB-PRYSM-11:EpochParticipation.UnmarshalJSONmayparseinvaliddata ResolvedinPR#12534.ThePrysmteamhasrevisedthesectionofthecodefordecoding Base64-encodedstrings.Intheoriginalimplementation,therewasanassumptionthatthe inputstringwasenclosedindoublequotes,andthesequoteswereremovedbefore decoding.However,intherevisedcode,theteamoptedtodirectlyconvertthebytesliceto astringandsubsequentlyemploythe strings.Trim() functiontoeliminateany surroundingdoublequotes. hasheye 77Prysmv3.2.2SecurityAssessment PUBLIC

TOB-PRYSM-12:Uint256.UnmarshalJSONmayparseinvaliddata ResolvedinPR#12540.ThePrysmteamupdatedthe UnmarshalJSON functioninboththe api/client/builder/types.go and beacon-chain/rpc/apimiddleware/structs_marshalling.go filestoverifythe presenceandcorrectnessofdoublequotesaroundtheJSONstringbyexaminingthefirst andlastcharactersoftheinputbyteslice.Ifthestringlacksproperdoublequotesoristoo short,itreturnsanerrormessageaccordingly. TOB-PRYSM-13:FailedassertionsintheFuzzExecutionPayloadfuzzingharness ResolvedinPR#12541.ThePrysmteamhasmadeupdatestoitsfuzzingexecutionpayload targettohandlecapellapayloadsandaddedlogictohandlethenilpayloadcaseinthe capellapayloadmarshalling.Theclientprovidedthefollowingcontextforthisfinding’sfix status: Wenowonlyfuzzexecutionpayloadsforcapellaandcomparethem.Gethaddsfieldsto

theirexecutabledata,sothereisnoeffectivewaytohandlethefuzztargetacross differentforks. TOB-PRYSM-14:TheJWTauthenticationdocssuggestgeneratingthesecretusing third-partywebsites ResolvedinPR#790.ThePrysmteamhasupdatedtheJWTauthenticationdocumentation byremovingthelinesuggestingtheuseofthird-partywebsitestogenerateauthentication secretvalues. TOB-PRYSM-15:Potentiallyinsufficientgossiptopicvalidation Unresolved.Theclientprovidedthefollowingcontextforthisfinding’sfixstatus: Wearenotabletotacklethisatthemomentbecauseitwouldrequirealargechangeto gossiptopicvalidationwhichwilltakeawhileandbringsitsownrisks.However,wedo eventuallyplantoaddressthisinthefuturewhenthereisbandwidthintheteam. hasheye 78Prysmv3.2.2SecurityAssessment PUBLIC

J.FixReviewStatusCategories Thefollowingtabledescribesthestatusesusedtoindicatewhetheranissuehasbeen sufficientlyaddressed. FixStatus StatusDescription UndeterminedThestatusoftheissuewasnotdeterminedduringthisengagement. UnresolvedTheissuepersistsandhasnotbeenresolved. PartiallyResolvedTheissuepersistsbuthasbeenpartiallyresolved. ResolvedTheissuehasbeensufficientlyresolved. hasheye 79Prysmv3.2.2SecurityAssessment PUBLIC