SORAMITSU

Polkaswap Exchange Penetration Test

Executive Summary

Tevora Threat Research Group

Delivered April 1, 2021

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

Purpose

The Polkaswap Exchange Pentration Test for SORAMITSU was conducted from January 21, 2021 to February 24, 2021 to help ensure SORAMITSU resources are 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 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 using supplied credentials
- Verification that only authorized services are exposed to the network perimeter
- Verification of network segmentation of non-privileged and privileged networks

Project Participation

The following SORAMITSU and Tevora personnel were directly involved in this assessment.

Sponsors	Ales Zivkovic, SORAMITSU
Participants	Iurii Vinogradov, SORAMITSU Pavel Golovkin, SORAMITSU Denis Nikoforov, SORAMITSU Clayton Riness, Tevora Krisopher Vaslik, Tevora Jade Talley, Tevora Brandon Pham, Tevora Kevin Dick, Tevora Sam Treece, Tevora

Scope

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

Web Application Penetration Test

The following items were considered in scope:

Source Code Repositories:

- sora2-substrate
- sora2-wallet-web
- sora2-substrate-js-library
- polkaswap-exchange-web
- sora2-ethereum-bridge-web

Web Applications:

- test.polkaswap.io/exchange/swap

Findings Overview

Web Application Penetration Test Results

Throughout all aspects this test, Tevora discovered up-to-date development practices across the Polkaswap environment. Static analysis appears to be used in parts of the development pipeline. Current standards were observed to be followed with encryption, encoding and handling data in transit. API endpoints properly sanitized data and returned error messages to bad data which were descriptive, but not descriptive enough to assist in the creation of malicious payloads. SORAMITSU required all transactions to be signed and used smart contracts to safeguard against theft or the forging of transactions.

Tevora performed this engagement against the public test instance of the Polkaswap Exchange web application while connected to the SORA-staging Testnet. Additionally, SORAMITSU provided Tevora with the source code for the application to use as an additional reference during dynamic analysis and to perform static analysis against. This application's primary focus is to act as a front end for users to import their Polkadot accounts and exchange certain types of cryptocurrency for others. There is functionality that allows users to view their current assets, manage their liquidity and set slippage tolerance. Additional features, all currently under development, allow users to generating new wallets with a seed phrase, viewing account activity and specifying which network to connect with. The source one can import existing wallets from is currently restricted to the polkadot.js browser extension. When an account is linked, the user can then add assets to track through a list or manually through a valid asset address. Once connected, the application opens a websocket connection to communicate with the Substrate based backend. This connection uses the json-rpc API to send and receive data about conversions, rates, status and other aspects of the greater network.

Tevora discovered small several areas where SORAMITSU could increase the security of the Polkaswap front end web application. An outdated installation of nginx was being used to host the application. While there isn't any currently known of public exploits for this version, there have been bugs and other minor issues fixed in newer version and the possibility of it containing unknown vulnerabilities is greater. Tevora also noticed the lack of several policies that could be implemented to even further increase the overall security. These include the lack of a Referrer policy, the lack of fully implemented HTTP Strict Transport Security (HSTS) and lack of a Content Security Policy (CSP) . In total it was observed that the application followed current recommendations and practices and most of the common issues seen across web applications were not present on the Polkaswap exchange.

The application has an API enabling the user and frontend web application to communicate with the Substrate based backend of Polkaswap. This API allows the user to perform actions such as query the network for the current heath of the network, get the assets currently registered with the network, details on those assets, check the current balance of a user and get exchange rate fees. A user can use the web applications provided by polkadot.js, write their own Javascript, use any of the language specific libraries provided by Polkadot or use the API endpoints to communicate directly with the Polkaswap API. For this engagement, Tevora used a combination of all available options to test and validate different parts of the API and data flows. This API allows for all aspects of communication with the Polkaswap substrate network and for that reason can perform a plethora of actions. All critical and sensitive actions require the user issuing the command to sign the transactions, which greatly increases the security of the platform and complexity of the attack needed to exploit. Tevora used a combination of fuzzing and manual modifications to these API calls with dangerous, incorrect and badly encoded data in attempt to perform unintended actions or receive unintended responses. Tevora modified API calls with data that would break the parsing of the json with the goal of causing unintended behaviors. Tevora also tested more obvious things like trying to "game the system" by attempting to get better exchange rates on token swaps and generate tokens "out of thin air". All calls appear to have been properly sanitized, checked and have appropriate error messages returned. Tevora was unable to perform any unintended actions through these exercises. Tevora did notice that when fuzzing assets_getAssetInfo, valid responses would be sent even if there was no asset registered to the fuzzed asset. However, this unexpected behavior did not lead to the discovery of anything that would be considered a vulnerability or configuration issue. Tevora was unable to discover any additional odd behavior or issues with the underlying network or network functionality.

SORAMITSU has also developed a bridge enabling users to transfer assets to and from the Ethereum bridge. This bridge uses Ethereum smart contracts, sidechain and Substrate multi-signature off-chain workers to transfer assets across networks. This bridge appears to be more of a work in progress than other parts of the Polkaswap ecosystem, however Tevora was able to test the general flows and functionality of this feature from one account. In addition to the aforementioned methods a user can integrate with the Polkaswap network, at the time of writing, this bridge requires the use of etherscan.io and the web3playground to perform certain aspects of the transactions to and from the sidechain. To transfer funds, a user submits a signed transaction which will then transfer the tokens to the side chain. From there the user can use the interface at etherscan.io to write a contract to move the tokens from the sidechain to Ethereum. The process is similar in reverse with, however the Ethereum to sidechain smart contract is currently written with using the tools at w3playground. Tevora inspected these data flows, all of which appeared designed and developed, and was unable to find any way that would allow a malicious entity to modify or takeover these transactions. Tevora also attempted to perform such actions as adding assets that the user does not own to the sidechain, transferring assets the user does not own to and from the sidechain, and attempting to take assets from the sidechain not destined for the user. The use of smart contracts and signed transactions greatly increases the overall security of this process and narrows down the possibilities an attacker has to exploit the network.

In addition to using the provided sourced code to assist in the dynamic analysis, Tevora manually reviewed and performed static analysis on the repositories. This analysis resulted in Tevora discovering no credentials, other authentication mechanisms or sensitive information that should not be present. Additionally, the static analysis resulted in a lack of findings for potential vulnerabilities or other areas where development should be focused on being improved. This helped strengthen the opinion of high-quality code and development workflow.

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 SORAMITSU. 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 an 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

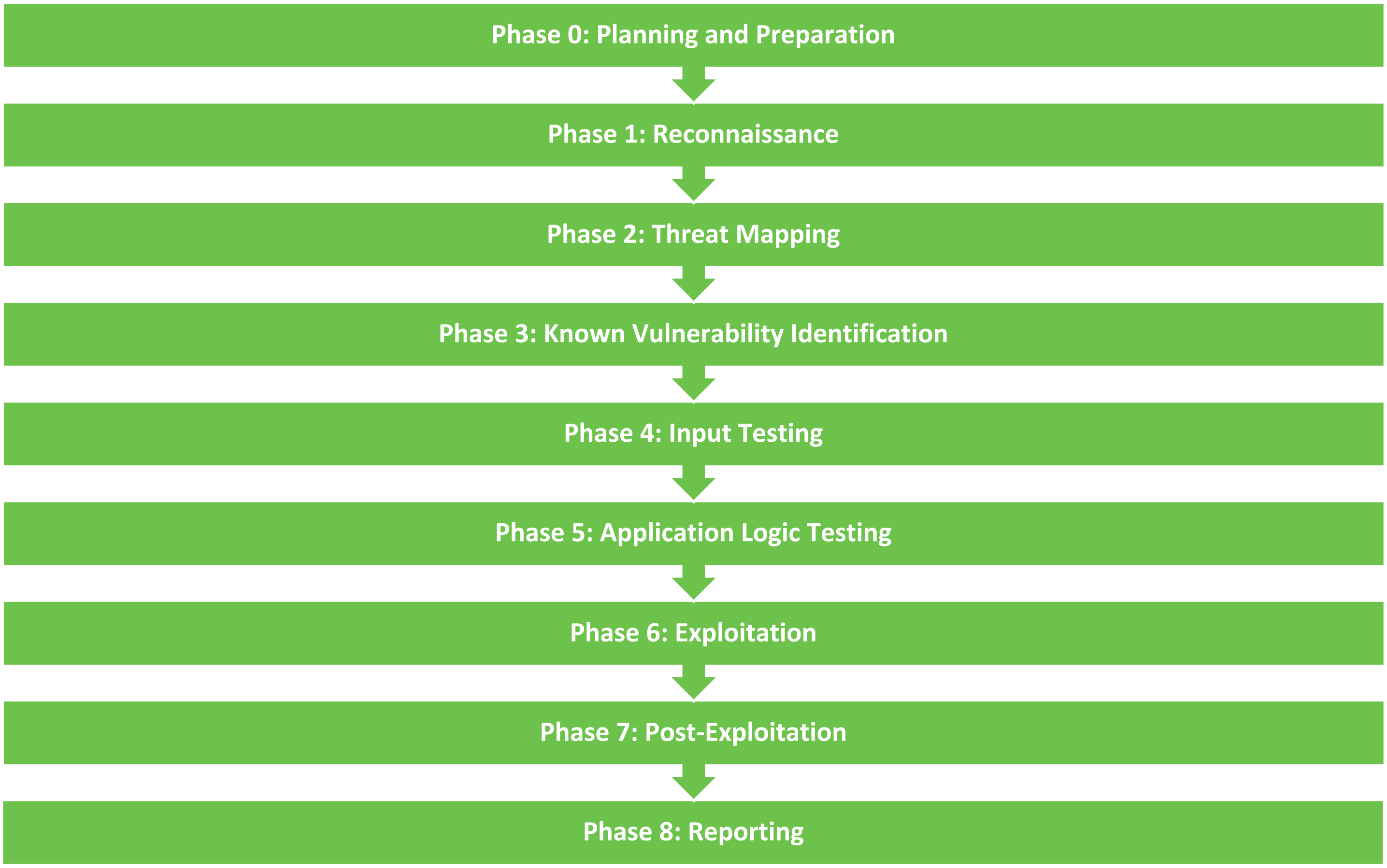
Penetration Testing Tools

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

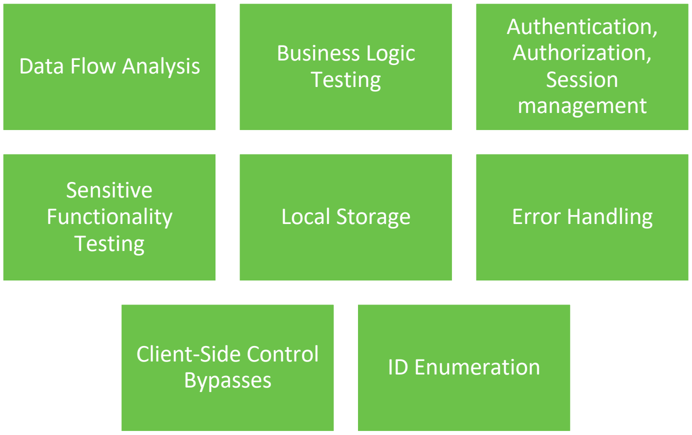
Any discovered input vulnerabilities are categorized, referenced against the threat map, and documented for use in the exploitation phase.

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 a primarily 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 as well as 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.

Appendix D: Assessor Profiles

Kevin Dick, Manager

Primary Role

Project Lead

Notable Accomplishments

Kevin’s threat research expertise includes network, web, and mobile application penetration testing, development of internal Tevora penetration testing and social engineering toolkits, malware analysis and incident response.

Kevin also develops and administers Tevora’s secured coding curriculum and training sessions.

Kevin’s solution advisory and implementation expertise spans Splunk, Okta, Dell Identity Manager, Intel Nitro SIEM, and FireMon and host of other solutions.

Certification and Training

Kevin obtained his bachelor’s degree in business information management from the University of California, Irvine. Kevin also is an Offensive Security Certified Professional (OSCP), a Splunk Certified Architect, a Certified Information Systems Security Professional (CISSP), and an ISO 27001:2013 lead auditor.

Project Role

Kevin is the project lead for this engagement. Responsibilities for this role include driving the entire delivery effort from on-site interviews to data collection and analysis, deliverable creation and review, and overseeing project status reporting.

Tenure

Kevin has been with Tevora since July 2012.

Sam Treece, Information Security Consultant

Primary Role	Technical Lead
Notable Accomplishments	Before joining Tevora, Sam worked as a security engineer providing a variety of security and development services to a wide variety of clients. Sam has spent over a decade working with embedded and mobile devices either by means of contributing to open-source projects to being one of several founders in custom firmware for Android devices used by millions of people where he had to manage multiple teams of people across the planet, write tools to make their development more streamlined and reviewed all code for potential vulnerabilities or other issues. For the last few years Sam primary focus has been on mobile applications, embedded devices and open-source intelligence (OSINT) gathering methods. Sometimes by assessing the applications or devices, building firmware for embedded devices to assist in another aspect of security engineering or researching and writing programs to assist in OSINT.
Project Role	Sam is the primary technical resource for penetration testing and vulnerability remediation. Responsibilities for this role include assisting and leading internal and external penetration tests that include network, web application, mobile, and cloud application penetration tests along with documentation of findings and client remediation of vulnerabilities.
Tenure	Sam has been with Tevora since March 2020.

TEVORA™

Go forward. We’ve got your back.

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

HYDRARISK

MODEL