Penetration Test

We help you build security into your software at every stage

Table of Contents

1 Overview

With the increasing popularity of smartphones, the app market is becoming ever more mature, and the development potential for mobile applications is huge. The mobile Internet industry is witnessing an explosive growth, with the number of developers having increased tenfold to hundreds to thousands. Millions of mobile applications are now available, and the number of Internet apps being developed every day is increasing at an exponential rate. The development and promotion of applications has become a huge market within the mobile Internet industry. App services cover a variety of industries such as hotels, beauty, automobile, health care, tourism, real estate, clothing, retail, entertainment and media. Currently, the main threats to the security of mobile applications are:

1. Android APK Trojans
2. Key information disclosure
3. APK repackaging
4. Process hijacking
5. Insecure data transmission
6. Hijacked keyboard input
7. Component vulnerabilities such as to Android's four components and WebView
8. Penetration test attacks on servers

The current security situation for mobile application development companies:

1. Most companies are unaware of the use of penetration testing.
2. Only a small number of information security companies have the ability to deliver excellent penetration test services.

2 Penetration Test Services Overview

A penetration test refers to the process in which safety engineers try to perfectly simulate vulnerability detection technologies and attack methods used by hackers in order to carry out in-depth exploration into the security of a target network/system/host/application and discover the weakest link in the system. It is a means to evaluate the security of a computer network system and can make customers aware of the problems encountered by their network.

A penetration test is a type of professional security service similar to the concept of "combat exercises" or "sand table exercises" used in the army. Through actual combat and exercise, it allows users to clearly understand the vulnerabilities of the current network and the possible impacts of such vulnerabilities, so that the necessary precautions may be adopted.

2.1 Values of penetration testing

Bangcle has accumulated many years of experience in mobile app vulnerabilities. Its professional security service team provides comprehensive penetration test services for mobile apps, as well as effective and feasible security solutions.

Main values:

1) To assist the user in finding the least secure parts in the organization, and to help the company effectively understand the initial tasks needed to reduce risks;
2) A complete and effective penetration test report helps to illustrate the current state of security, thereby enhancing the awareness of information security and even improving the organization's security budget;
3) Information security is an integral project. Penetration testing helps all members in an organization to be more aware how their jobs may improve or reduce risks, which helps improve internal security.

Of course, a penetration test does not guarantee to discover all vulnerabilities in a target network, therefore we should not place too much one-sided emphasis on its importance.
As a result, penetration testing has even become a measure used by the customer to test a security company.

2.2 Mission of penetration test services

Sincere and thoughtful: Think what customer wants, be eager to help, take the initiative to meet needs, show concern, be genuine, kind, sincere in communication and responsible for the details.
Professional and efficient: Guaranteed professional penetration testing, standardized service, measurable quality of service, timely customer response, prompt solutions to problems, reliable tools, continual improvement and effectiveness of service and a professional and efficient service.

2.3 Objectives of penetration test services

Discover risks and vulnerabilities: According to industry standards and professional practices, identify the threat risk and discover all security vulnerabilities in the tested system.

Fix app security vulnerabilities: Provide professional solutions to security vulnerabilities and fix all items at risk.

Avoid subsequent business and regulatory risks: Provide aversion measures for professional business logic security and regulatory risks to prevent business and regulatory risks affecting the normal implementation and use of the system.

2.4 Commitment of penetration test services

- To meet regulatory requirements for mobile apps
- To meet technical requirements for mobile payment in the financial industry
- To meet relevant international security standards for information technology
- To remove obvious technical defects that can be attacked

3 Penetration Test Techniques

For a long time, penetration testing has been draped in mystery. The main reasons for this are that operators of penetration tests are not only skilled at using a variety of tools, but penetration testing can also apply some unique methods to break the defense of a network system and obtain access.

Of course, just like reasoning of Sherlock's Holmes, these seemingly impossible tasks can, in fact, be achieved through thorough skills training and the use of reverse and divergent thinking.

The various phases in a penetration test are introduced below (similar to stages of a hack). Tool may be used throughout, but the essence of application lies in dedication and breakthrough in thinking.

3.1 Penetration test preparatory phase

Basic information acquisition:
The APK is unpacked using apktool or other tools to access information such as the package name, signature, resource files, and the dex file. Basic information and other useful information of the tested APK is understood and used for further preparations for penetration testing.

WhatWeb and other Web fingerprint identification tools are used to identify the site and create a program to use, including the system type, JavaScript library, a Web server, embedded devices, and other Web fingerprint information.

Nmap carries out port scanning of the website and determines the type of operating system.

3.2 Penetration test phase

The attacks based on the APK app, APK Server API are directed at weak spots in network applications of B/S or C/S structures. Commonly seen attacks are types such as command execution, privilege escalation, SQL injection attacks and cross-site scripting attacks. The main functions involve the following points:

Security penetration test of APK components: Includes penetrating testing on the security of four major components of the APK, namely Activity, Service, Content Provider and Broadcast Receiver, as well as WebView, Fragment and other components. Component security issues may include denial of service, command injection and access control.

Security penetration test of APK Server: Includes security vulnerabilities relating to OWASP TOP 10.

Business penetration test: Achieves the effect of an attack through targeted penetration testing according to the different businesses of each product.

3.3 Penetration test later phase

The shell of the mobile app system is obtained to eliminate traces of the penetration test and achieve long-term access to system permissions. This phase mainly includes two aspects:

- Obtaining GetShell from the mobile app system
- Deleting the system log to avoid subsequent tracking

3.4 Other techniques

Some of the methods listed here may greatly affect the user's network (e.g. service interruption); some are closely related to security management (not only from technical considerations); and some only work live, therefore they are rarely used by penetration testers under normal circumstances. However, this is determined according to the specific needs of the customer.

- DoS&DDoS
- Social engineering approach

4 Penetration Test Cases

4.1 Android client

4.1.1 Tools

The tools used include, but are not limited to, the tools listed in the following table:

No	Tool name	Version information	Description
1	apktool	Version 2.0.0	APK decompiler/recompiler
2	ActivityHijack	Version 1.0 (independently-developed)	Interface hijacking testing tool
3	gdb	Version 6.7	Dynamic debugging tool for dynamic debugging of app programs
4	DDMS	Version 22.0.1	Android debug monitor with integrated tools such as the Logcat viewer used to debug log messages, screenshot tools and memory analysis tools
5	AXMLPrinter2	Version 1.0	AXMLPrinter2 is a tool used to decompile and decrypt Android XML files; it can decompile Android binary XML files into plain text form
6	RootExplorer	Version 3.3.4	A highly-authorized file manager used to view and modify any system file after obtaining root permission
7	MemSpector	Version 2.0	Accesses the memory of any application process, and views and modifies memory data searches after obtaining root permission
8	Wireshark	Version 1.12.4.0	A network packet analysis tool
9	Fiddler	Version 4.6.0.2	Fiddler is a HTTP debugging proxy tool, which is able to record and inspect all network traffic between your computer and the network, set breakpoints, and view all incoming or outgoing data
10	signapk	Version 1.0	A signature tool used for the APK

			installation package for Android applications
11	Dex2jar	Version 0.0.9.15	An APK decompiler, which can obtain Java bytecode after decompiling APKs
12	Jd-gui	Version 1.1.0	A Java decompiler, which can decompile Java bytecode into Java source code
13	7-ZIP	Version 15.5.0.0	File compression/decompression software
14	Uedit32	Version 19.10.0.12	Text viewer/editor
15	adb	Version 1.0.31	Control and management of an Android emulator or device, with app installation and uninstallation functions
16	drozer	Version2.3.4	Android app security testing tool

4.1.2 Check points

4.1.2.1 Binary code protection

This test is carried out on the binary file after the release of the mobile app with the aim of checking whether the application can effectively prevent reverse engineering or decompilation. It includes the application code, resource files, script files and business-related sensitive documents, and ensures source integrity after the app is released. At the same time, it is also used to detect whether the app is able to resist runtime attacks (e.g. binary code injection, memory search and memory read) when operating in an insecure environment (e.g. root, jailbreak, etc.) in order to ensure that security of business data of the app can still be guaranteed.

Test item	Level
DEX (classes.dex files) code encryption protection	Basic
DEX (classes.dex files) dynamic code debug test	Intermediate
Native library (so files) code encryption protection	Basic
Native library (so files) dynamic debug test	Intermediate
Native library (so files) injection test	Intermediate
Script code (JavaScript, Lua, Python, etc.) encryption protection	Basic
Encryption protection of sensitive resource files	Basic
Encryption protection of sensitive business files (certificates, configuration, etc.)	Basic
App signature and publication standard	Basic
Runtime signature check	Intermediate
Runtime file integrity check	Intermediate
Root environmental test and user prompts	Advanced

Xposed Framework injection test	Advanced
Repackaging test (static file tampering)	Basic

4.1.2.2 Storage security of client data

This test checks whether the mobile app provides effective protection for data generated during runtime, and ensures private data and authentication information cannot be obtained by a third party when the device operates in insecure environments (e.g. Root, phishing apps, Trojan viruses, and jailbreak) or is accidentally lost.

Test item	Level
Plain text storage of sensitive information in SharedPreferences files	Basic
Permission settings error for SharedPreferences files	Basic
Plain text storage of sensitive information in external memory (SD card)	Basic
Plain text storage of sensitive information in SQLite database	Basic
Plain text storage of log messages in external memory (SD card)	Basic
Storage and loading of DEX files in external memory (SD card)	Basic
Storage and loading of SO files in external memory (SD card)	Basic
Loading unauthorized files in external memory (SD card)	Basic

4.1.2.3 Data transmission protection

This test checks the security of data when transmitting network data between a mobile app and a business server, other device or terminal. It aims to analyze whether the communication between the app and the server is able to resist man-in-the-middle (MITM) attacks, and whether there is any monitoring, interception, tampering or replay by third-party programs.

Test item	Level
Exchange of core business data via HTTP	Advanced
Unverified HTTPS certificate	Advanced
Ignored certificate domain check	Intermediate
Plain text network transmission of core business data	Advanced
Plain text data transmission over other channels	Intermediate
Lack of integrity for core business data requests	Advanced

4.1.2.4 Encryption algorithms and password security

This test checks whether the mobile app conforms to relevant criteria and standards when using cryptography-related functions. It includes the use of encryption and hash algorithms, and ensures that all encryption algorithms used by the mobile app during data storage and transmission can prevent attackers from decrypting the data within a limited period of time. At the same time, this

test also focuses on issues related to generation, use, transmission and storage of encryption keys, and ensures that the app follows the best security practices to ensure crypto-key security.

Test item	Level
Use of insecure encryption algorithms	Advanced
Use of insecure hash algorithms	Advanced
Improper use of encryption algorithms (e.g. ECB mode)	Advanced
Use of fixed and hard-coded crypto-keys	Intermediate
Crypto-key length does not conform to specifications	Basic

4.1.2.5 Security of cross-process interaction

This test checks the security of components that may be accessed by other processes when the Android app is running. These components include Activity, Service, BroadcastReceiver, ContentProvider as well as TCP/UDP servers running in the background. These components that may be accessed by other processes may present problems such as denial of service (DoS), logic errors and privilege escalation when accepting external transfer parameters. This test ensures that the app is protected from malicious third-party programs when running.

Test item	Level
DoS attack on exported Activity component	Basic
DoS attack on exported Service component	Basic
DoS attack on exported Broadcast Receiver component	Basic
DoS attack on other exported components	Basic
Data breach from exported Content Provider component	Basic
SQL injection in exported Content Provider component	Advanced
Local business logic vulnerabilities generated by exported Activity component	Basic
Local business logic vulnerabilities generated by exported Service component	Basic
Local business logic vulnerabilities generated by exported Broadcast Receiver component	Basic
Business logic vulnerabilities generated by other exported components	Basic

4.1.2.6 Security specifications for Android apps

This test addresses common security issues during the development of Android apps and focuses on whether the mobile app follows the best security practices.

Test item	Level
Logcat output log contains sensitive information	Basic
Backup data flag enabled	Basic
DEX code debugging flag enabled	Basic
Incorrect use of WebView component	Intermediate

Hijacking and redirection of automatic app upgrade	Advanced
Sensitive Activity preventing screen capture	Basic
Using built-in virtual keyboard for sensitive input	Intermediate

4.2 iOS client

4.2.1 Tools

The tools used insclude, but are not limited to, the tools listed in the following table:

No	Tool name	Version information	Description
1	IDA pro	Version 6.6	An interactive decompiler, which supports dynamic debugging and static decompilation of executable files on multiple platforms
2	iTools	Version 3.2	A phone management software that supports Android and iOS
3	gikdbg	Version 1.2	A dynamic debugger used to carry out dynamic debugging of iOS app programs
4	7-ZIP	Verison 15.5.0.0	File compression/decompression software
5	Uedit32	Version 19.10.0.12	Text viewer/editor
6	Wireshark	Version 1.12.4.0	A network packet analysis tool
7	Fiddler	Version 4.6.0.2	Fiddler is a HTTP debugging proxy tool, which is able to record and inspect all network traffic between your computer and the network, set breakpoints, and view all incoming or outgoing data

4.2.2 Check points

4.2.2.1 Binary code protection

This test is carried out on the binary file after the release of the mobile app with the aim of checking whether the application can effectively prevent reverse engineering or decompilation. It includes the application code, resource files, script files and business-related sensitive documents, and ensures source integrity after the app is released. At the same time, it is also used to detect whether the app is able to resist runtime attacks (e.g. binary code injection, memory search and memory read) when operating in an insecure environment (e.g. root, jailbreak, etc.) in order to ensure that security of business data of the app can still be guaranteed.

Test item	Level
Sensitivity function obfuscation	Basic
Script code (JavaScript, Lua, Python, etc.) encryption protection	Basic
Encryption protection of sensitive resource files	Basic
Business-sensitive files (certificates, configuration, etc.)	Basic
Jailbreak environmental test and user prompt	Advanced

4.2.2.2 Storage security of client data

This test checks whether the mobile app provides effective protection for data generated during runtime, and ensures private data and authentication information cannot be obtained by a third party when the device operates in insecure environments (e.g. phishing apps, Trojan viruses, and jailbreak) or is accidentally lost.

Test item	Level
Plain text storage of sensitive information in plist files	Basic
Plain text storage of sensitive information in SQLite database	Basic

4.2.2.3 Data transmission protection

This test checks the security of data when transmitting network data between a mobile app and a business server, other device or terminal. It aims to analyze whether the communication between the app and the server is able to resist man-in-the-middle (MITM) attacks, and whether there is any monitoring, interception, tampering or replay by third-party programs.

Test item	Level
Exchange of core business data via HTTP	Advanced
Unverified HTTPS certificate	Advanced
Ignored certificate domain check	Intermediate
Plain text network transmission of core business data	Advanced
Plain text data transmission over other channels	Intermediate
Lack of integrity for core business data requests	Advanced

4.2.2.4 Encryption algorithms and password security

This test checks whether the mobile app conforms to relevant criteria and standards when using cryptography-related functions. It includes the use of encryption and hash algorithms, and ensures that all encryption algorithms used by the mobile app during data storage and transmission can prevent attackers from decrypting the data within a limited period of time. At the same time, this test also focuses on issues related to generation, use, transmission and storage of encryption keys, and ensures that the app follows the best security practices to ensure crypto-key security.

Test item	Level
Use of insecure encryption algorithms	Advanced
Use of insecure hash algorithms	Advanced
Improper use of encryption algorithms (e.g. ECB mode)	Advanced
Use of fixed and hard-coded crypto-keys	Intermediate
Crypto-key length does not conform to specifications	Basic

4.2.2.5 Security specifications for iOS apps

This test addresses common security issues during the development of iOS apps and focuses on whether the mobile app follows the best security practices.

Test item	Level
Sensitive information in debug log output	Basic
No obfuscation of background cache on sensitive interface	Intermediate

4.3 Web interface and business logic vulnerabilities

4.3.1 Check points

4.3.1.1 Injection attacks

Injection attacks occur when untrusted data is sent to the interpreter as part of a command or query. The malicious data sent by the attacker can trick the interpreter into executing unintended commands or accessing unauthorized data. Take SQL injection for example, if the server does not effectively filter the data submitted by the client, this may result in the SQL statement executed by the server being maliciously tampered with or the original SQL structure being destroyed, ultimately tricking the server into executing malicious SQL commands. Attackers using SQL infection may obtain a variety of information in the database (e.g. administrator password and user login account and password), thereby removing all data within the database.

Test item	Level
SQL injection	Advanced
XXE injection attack	Advanced
Command injection attack	Advanced
XML injection attack	Basic

4.3.1.2 Identity authentication and session management

An account system mainly consists of four functions: user registration, user login, session

management, and password reset. If the server fails to take adequate security measures when dealing with business logic, this will result in a series of serious consequences such as user password disclosure, malicious user account login and registered spam accounts.

Test item	Level
Brute force attack on password	Intermediate
Brute force attack on user name using weak password	Intermediate
Arbitrary password reset	Intermediate
Third-party login defect	Intermediate
Arbitrary user login	Advanced
Malicious registration	Advanced
Defect of bypassing SMS verification code	Advanced
SMS interface replay attack	Advanced
Login interface replay attack	Advanced
Payment interface replay attack	Advanced
Fixed session token	Advanced
Insecure session delivery	Intermediate
Failure to exit session	Intermediate
No set session validity period	Basic

4.3.1.3 Information disclosure

When there are configuration or other logic defects in the target system, an attacker may access the server's configuration file or cause the server to generate an error by constructing a special data packet, thereby obtaining sensitive server or user information.

Test item	Level
Too much unnecessary sensitive data returned by the server	Basic
No obfuscation carried out in the data packet returned by the server	Basic
Server error message	Basic

4.3.1.4 Business defects

Apps include businesses such as personal information CRUD, commodity transactions, and multi-factor authentication. In dealing with these businesses, the server may overly trust the data submitted by the client, or the client may overly trust the data returned by the server. An attacker can tamper with the data to trick the server and the target destination on the client's side, which may result in problems such as information disclosure and malicious transactions.

Test item	Test result
Sequential execution defect	Advanced
Race conditions	Advanced
Logic vulnerabilities	Advanced

4.3.1.5 Missing function level access control

If an app does not implement adequate account access control, ordinary users can access other account data, or possess authorized access to the account, which may result in malicious viewing, modification or deletion of account data.

Test item	Level
Horizontal privilege escalation	Intermediate
Vertical privilege escalation	Intermediate

4.3.1.6 Access control vulnerabilities

If the server does not implement adequate access control, an attacker may access system file and server configuration files, and even upload Trojan files to gain control of the server.

Test item	Level
File upload vulnerability	Intermediate
File download vulnerability	Intermediate
File read vulnerability	Intermediate
Directory traversal vulnerability	Advanced

4.3.1.7 Cross-site scripting

If a page does not effectively filter or transform data submitted by the user, an attacker may insert and execute malicious codes into the page, which may impact or cause loss to the user.

Test item	Level
Cross-site scripting (XSS) attacks	Advanced

4.3.1.8 Using components with known vulnerabilities

Defects in the framework components or defects with the developer's code itself, timely patch updates may not be received during security maintenance. This may result in attackers exploiting these vulnerabilities to execute remote commands or access sensitive information.

Test item	Level
Strusts2 vulnerability	Advanced
Java deserialization vulnerability	Advanced
JBoss vulnerability	Advanced
Tomcat vulnerability	Advanced

WebLogic vulnerability	Advanced

5 Attack Paths

Different techniques are required for different test objectives. The following gives a brief description of the techniques that may be used for different locations and attack paths.

5.1 Client penetration test of app

A client security penetration test of an app is to perform a penetration test on an APK. Release versions of APKs are usually protected by a shell and obfuscated code, therefore three stages of unpacking, de-shelling and decompilation are needed to be able to conduct a thorough security penetration test.

Please refer to the security penetration checks in the appendix for specific APK penetration test items.

5.2 Server penetration test of app

A server penetration test of an app is to conduct a penetration test by adopting an external network. An external network test refers to the behavior simulated by an external attacker with no knowledge of the internal state, and in which the penetration tester lies completely in an external network (e.g. dial-up, ADSL or external optical fiber). It mainly consists of the following aspects:

1) Testing and evasion of firewalls;
2) Testing the security of Web and other open applications;

Please refer to the security penetration checks in the appendix for specific app server penetration test items.

6 Penetration Test Service Process

6.1 Written consent from client

Legitimacy, which refers to the written authorization from a client and indicates approval of the implementation plan, is a necessary condition for implementing a penetration test. The implementation method, time, personnel, tools and other specific penetration test plans must first be submitted to the client, who then issue a corresponding written statement of entrustment and authorization.

Customers should be fully aware of the details and risks of the penetration test, and all processes are under their control. This is the difference between professional penetration test services and hacks.

6.2 Formulation of implementation plan

Penetration test services by Bangcle mainly consist of 5 steps, as shown in the diagram below:

- Product familiarity: Being familiar with the product and business scenarios according to the product manual or other methods.

- Analysis of business data flow: Drawing a data flow diagram according to each business scenario and highlighting key information assets for each business scenario generated by the data flow diagram.

- Threat modeling: Conducting security threat modeling according to the business data flow and relevant information assets, and integrating the security hazard model STRIDE.

- Penetration test plan and execution: Implementing the penetration test plan for the mobile app according to STRIDE threat modeling.

- Output of penetration report: Outputting a penetration test report and suggesting solutions according to the penetration test results.

6.3 Collection of product information

Collecting information is the premise for each step of a penetration attack, with which a targeted simulated attack plan can be formulated to improve the success rate. It also effectively reduces the adverse effects of the test on the normal operation of the system.

Methods used to collect information include PingSweep, DNSSweep, DNSzone transfer, operating system fingerprint identification, application discrimination, account scanning and configuration discrimination. Common tools for collecting information include business network vulnerability scanning software (e.g Rapid7) and free security testing tools (e.g. NMAP and NESSUS). Many features built in to the operating system (e.g. TELNET, NSLOOKUP and IE) can also be used as an effective tool for collecting information.

6.4 Formulation of penetration test plan

A penetration test plan is formulated according to chapter 5.2 business flow analysis, key asset analysis, and key penetration test scenarios presented in the security threat modeling steps. The main items of a penetration test are as shown below:

1. Conventional penetration testing of mobile app APKs consist of the following aspects:
   - Improper use of platform
   - Insecure data storage
   - Insecure communication
   - Insecure identity authentication
   - Insufficient encryption
   - Insecure authorization
   - Client code quality issues
   - Code tampering
   - Reverse engineering
   - Irrelevant functions

2. Business penetration testing of mobile apps consist of the following aspects:
   - Injections, including SQL, XML, command, and code injections
   - Invalid identity authentication and session management
   - Cross-site scripting (XSS)

- Insecure object direct references
- Security configuration errors
- Sensitive information disclosure
- Missing function level access control
- Cross-site request forgery (CSRF)
- Using components with known vulnerabilities
- Unverified redirectioning and forwarding

6.5 Penetration test execution

The following possibilities may present themselves by collecting and analyzing preliminary information and formulating a penetration test plan:

- There is a significant risk of security vulnerabilities in the mobile app, and penetration testing can directly control the target mobile app.

- There is no significant risk of remote security vulnerabilities in the target mobile app, but ordinary user permissions may be obtained, which can be used to further collect information about the target system. You can then try hard to access superuser permissions, collect information on the target host, and seek opportunities to enhance local permissions.

- Business logic vulnerabilities exist in the mobile app. For example, this kind of vulnerability can result in business attacks such as business privilege escalation and logic bypassing of business authentication.

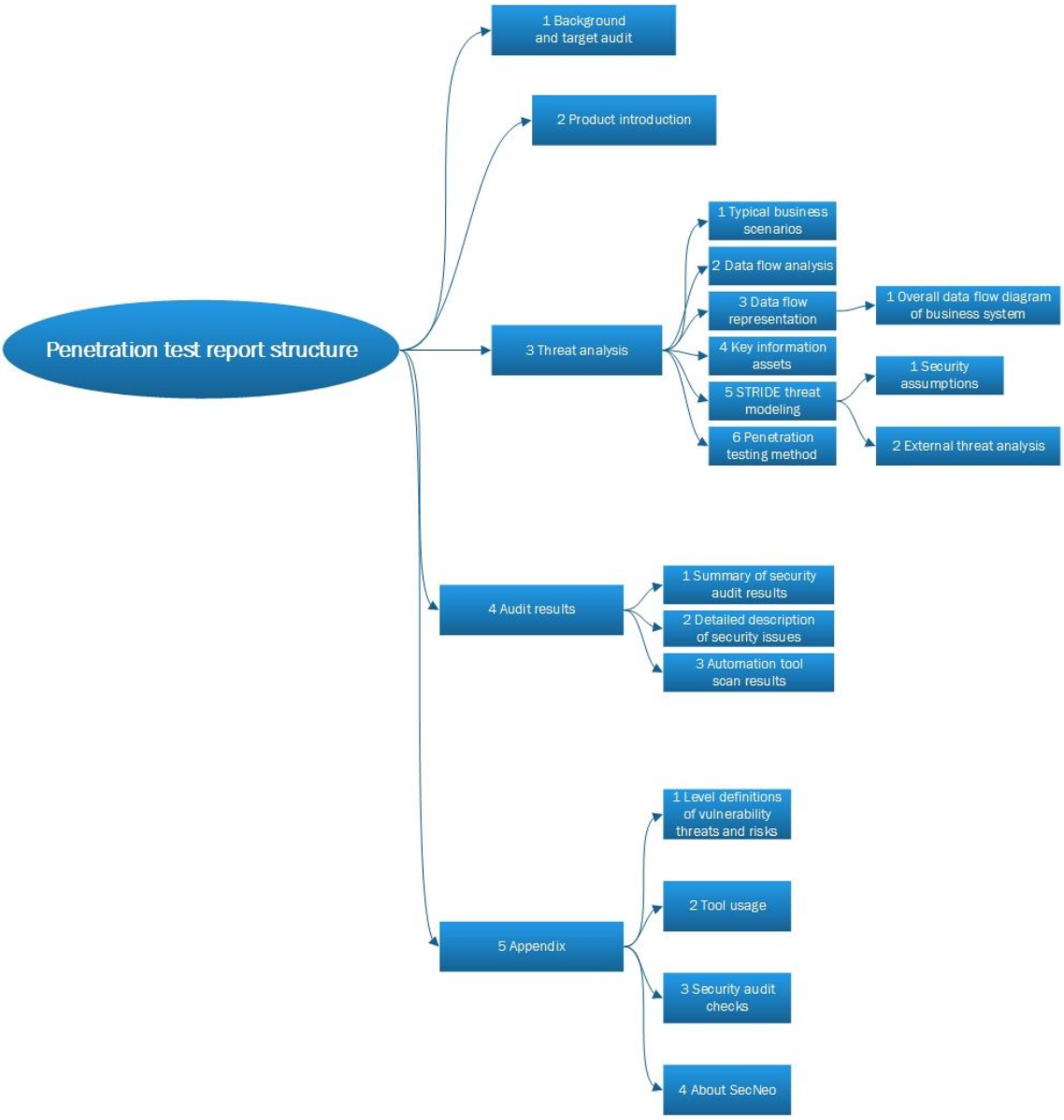
Continual information collection and analysis, system privilege escalation, and the result of business attacks form the entire penetration test process.

6.6 Penetration test report generation

After the penetration test is complete, the tester will provide a penetration test report. The report will provide a detailed description of the data and information obtained during the test, and will include a detailed record of all operations within the entire penetration test.

7 Penetration Test Report Compilation

A valuable penetration test report can help customers quickly locate the weak links in the organization, and avoid potential risks at the lowest cost. Accuracy and conciseness is key to a penetration report.



Some key points are highlighted in bold. Conclusions to penetration tests should be concise, clear and facilitate customers or developers in quickly understanding the crux of the problem. Operations in the pre-attack phase are the focus of most companies, as they not only want to know how they may be attacked, but also what methods were used by the penetration testers, what type of attacks they face and where they are secure. Specific operations in the attack phase

are undoubtedly the highlight of the report. Evidence only needs to be listed simply to highlight the main issues of the report, but solutions must be detailed.