PLETHORA
digital technologies

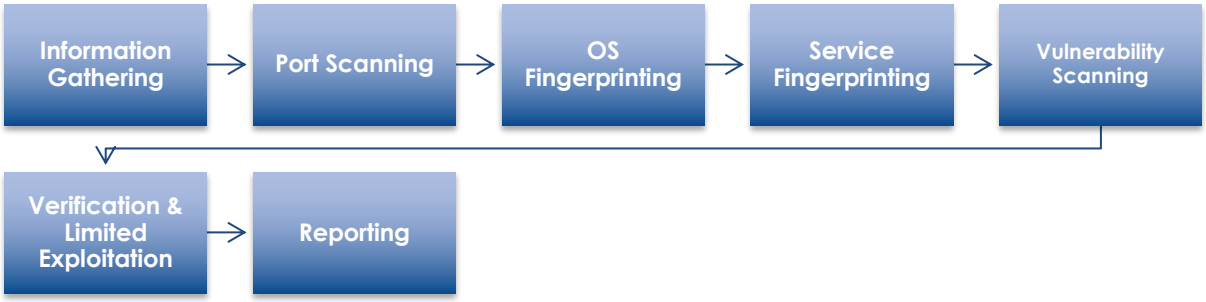
Network, Web and Mobile Applications Penetration and Vulnerability Assessment Test Approach Paper & Audit Scope Questionnaire

Unit 458 Kyalami, Maple Avenue,
Midrand, Johannesburg,
Gauteng, South Africa

+27 760 564 128
info@plethoradigital.tech
www.plethoradigital.tech

Lead Contact
Packson Ntanga

Methodology Overview (Network Security)



- **Information Gathering:** Collect information about the target network without running any intrusive tests, and use the data at later stage to exploit vulnerabilities
- **Port Scanning:**
  - Port scanning is the invasive probing of system ports on the transport level. This parameter is to enumerate live or accessible Internet services as well as penetrating the firewall to find additional live systems. Various tools and techniques are used to perform testing in stealthy mode. This is to prevent alerts and bypass IDS, IPS and Firewall
  - The expected results include open, closed or filtered ports, IP addresses of live systems, List of discovered tunnelled and encapsulated protocols, list of discovered routing protocols supported, active services, network map
  - Techniques used include TCP half, full; FIN, ACK and stealth scan of systems. Firewalk the network for detecting filtered ports. Packet fragmentation and session splicing techniques are used for stealthily passing through firewalls and IDS systems; SNMP querying is carried out for additional network information
- **OS Fingerprinting:**
  - System fingerprinting will be done for active probing of system for responses that can distinguish unique systems to operating system and version level. Expected result includes OS type and patch level.
  - Techniques used include Identification of operating system version and patch level based on responses to customized TCP/UDP packets, ICMP responses, OS banners and TCP sequence numbers
- **Services Fingerprinting:**
  - This is the active examination of the application listening behind the service. In certain cases, more than one application exists behind a service where one application is the listener and the others are considered components of the listening application. A good example of this is PERL installed for use in a Web application. In that case the listening service is the HTTP daemon and the component is PERL.
  - Expected result includes service type and patch level. Application identification is carried out by capturing application banners, responses to custom queries, analysis of websites for internal links, platform information and protocol behaviour.
- **Vulnerability Scanning:**

  - This exercise involves use of automated tools to test for vulnerabilities to determine existing holes and system patch level. Expected results include list of system vulnerabilities, type of application or service by vulnerability, patch levels of systems and applications, list of possible denial of service vulnerabilities.
  - Techniques used include comparison of system information collected with public security databases to determine system security holes, searching online databases and mailing lists specific to the systems being tested.
- **Verification & Limited Exploitation:**
  - In this step vulnerabilities identified in the previous steps are manually verified. Post verification exploits are divided into Harmless and Harmful exploits. Where ever possible controlled exploitation if vulnerabilities identified as not harmful are carried out. Harmful exploits are performed post confirmation from the customer.
  - Expected results include demonstration of exploited vulnerabilities such as Password Cracking or Buffer Overflow to gain unauthorized access or demonstrating administrative access to the compromised system.
- **Reporting:**
  - A report detailing all the discovered vulnerabilities in the system along with specific solutions to mitigate each risk is provided in the report. Based on the discovered risks in the IT infrastructure, we will recommend practical solutions and develop an implementation roadmap for strengthening security. This will involve patch recommendation, suggestions on improving practices & policies and options on security products for controlling the discovered risks. Assessment would look at risks from Internet, internal and through access points including RAS servers.

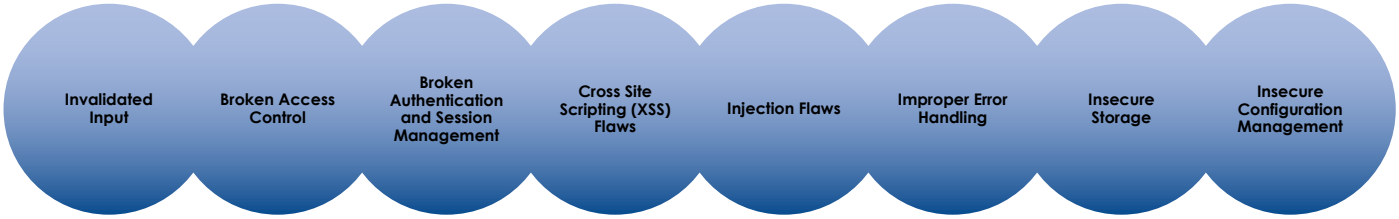
Methodology Overview (Web Applications)

Our methodology includes the use of an automated security-scanning tool supported by manual hacking and enumeration procedures to detect and exploit vulnerabilities:

- All prerequisite information about the systems and networks is gathered to allow an Impact Analysis to be completed;
- Realistic threats which are relevant to the organization are defined and agreed upon with application owners or custodians;
- Appropriate tests targeting key mission critical applications, application modules or access methods are identified; and
- Limitations to the proposed security assessment are determined (e.g., DoS testing, Physical Security Reviews etc.)
- **Uninformed outsider testing (Blackbox Approach) –** Emulates attacks from individuals with no significant knowledge of the application; testing is performed without credentials to the client application.
- **Informed Insider Testing –** Emulates attacks from employees, customers, or contractors with legitimate access to the application; testing is performed with credentials to the application.

Combining proprietary/industry approved tools & advanced manual methodologies, mission critical web applications will be assessed for identification & exploitation as per following criterions:

- Access to Sensitive Data
- Administrative Access
- Authentication
- Backdoors
- Cookie Poisoning
- Forceful Browsing
- Hidden Field Manipulation
- Infrastructure Misconfigurations
- Informational Access
- Application Misconfiguration
- Input Validation
- Known Vulnerabilities
- Session Management
- SQL Injection for Database
- Stealth Commanding
- Parameter Manipulation
- Third Party Misconfigurations
- Platform Misconfigurations



- **Access to Sensitive Data** – Sensitive files such as website backup files or configuration files containing passwords may be present on the web server and accessible to unauthorized individuals
- **Administrative Access** – Default or easily guessable passwords may be used to gain access to the administrative section of the website Authentication – Obtain unauthorized access to data or functionality due to authentication weakness
- **Backdoors** – Identify malicious entry paths that provide administrative access into the application or system (identified through manual methods).
- **Cookie Poisoning** – Change cookie data to access sensitive information or impersonate another user (identified through manual methods).
- **Forceful Browsing** – Access a web page or directory directly that should only be reached as an authenticated user (identified through manual and automated methods).
- **Hidden Field Manipulation** – Modify hidden field values to manipulate web application functionality or access sensitive data (identified through manual methods).
- **Informational** – View unauthorized data, such as personally identifiable information of another user (identified through manual methods).
- **Infrastructure/Platform Misconfigurations** – Web servers may be inadvertently misconfigured to permit anonymous authoring through FrontPage Server Extensions, file transfers with WebDAV, directory indexing and browsing, etc.; (identified through manual and automated methods);
- **Input Validation** – Input unauthorized types or amounts of data to determine if the application responds erratically (identified through manual and automated methods).
- **Known Vulnerabilities** – Identify known vulnerabilities present in the web application (identified through manual and automated methods).
- **Misconfigurations** – Identify and exploit system configuration settings that provide excessive access to application configuration, data, or functionality (identified through manual methods).
- **Parameter Manipulation** – Manipulate parameters by using special characters or "illegal" input such as SQL or JavaScript code to cause the web application to behave in an unauthorized or inappropriate manner (identified through manual and automated methods).
- **Session Management** – Compromise session management technology to hijack another session (identified through manual and automated methods).
- **SQL Injection** – Insert SQL code into form fields to bypass authentication or execute SQL statements directly on the database (identified through manual and automated methods).
- **Stealth Commanding** – Insert code in form fields to take control of an application or its host operating system (identified through manual and automated methods).
- **Third-Party Misconfigurations** – Exploit configuration errors in third party applications such as database servers (identified through manual and automated methods).

Methodology Overview (Mobile Application Security Assessment)

The world is becoming smarter everyday with smarter mobile technology. There is an increased demand for smart applications especially in the area of Banking and Retail sector. The increasing reliance on these applications has given rise to major security issues. While most enterprises focus on releasing mobile applications in a short span of time to keep up with the competition, security considerations are often overlooked. Compared to desktop or web applications, mobile applications are difficult to test for security since they run on devices that are not managed by the enterprise which stores tremendous amount of personal, commercial and financial data that attracts both targeted and mass-scale attacks.

Security leaks and confidential data disclosure from web and mobile apps are quite common today. With the increasing number of technologically rich mobile applications hitting the market, mobile phones have become the new target for hackers.

The Mobile Application Threat Landscape

Mobile devices and apps are becoming ubiquitous to both personal and professional lives, allowing for near anytime access to critical information. As a result, mobile device operating systems and applications are immensely vulnerable to security risks. It is crucial to identify and fix these risks at regular intervals.

- **Application-based threats**
- **Web-based or data-stealing threats**
- **Network-based threats**
- **Physical threats**

Processes and methodology behind our mobile app vulnerability testing services are based on the well-known standards and check-lists described by OWASP Mobile Security Project and Cloud Security Alliance (CSA) Mobile Application Security Initiative.

Mobile Application Security Testing Overview

Our Mobile Application Security testing services enables developers to focus on identifying and fixing security issues. We help enterprises gain security assurance for every mobile application that is being developed. Our security testing services are focused at identifying security risks under the four broad security threat areas.

Our Mobile application security consultants conduct a comprehensive security test on mobile applications, using an established and proven testing methodology that leverages off-the-shelf tools, automation scripts for various platforms that are capable of identifying threats specific to the application – even those related to its business logic, rules and processes.

A detailed actionable report(s) will be delivered with in-depth explanations on vulnerabilities, specifically indicating vulnerabilities in application feature and code along with a possible remediation (where possible).

Our "Post- remediation" security test can quickly confirm or report if all the security issues reported have been taken care of.

Our mobile application security testing solution ensures apps are secure before they go live and every new version undergoes rigorous security testing against a 12-point stringent certification criterion that maps to **OWASP Mobile Top 10, SANS Top 25, and other regulatory standards like PCI-DSS**.

Achieving compliance to security standards like OWASP mobile top 10 is a key factor to gaining your customer trust for your mobile applications.

Assessment types

We offer 2 types of security assessments for mobile applications, both of these lead to security certification. Depending on the availability of application, app user credentials and source code a particular type of assessment can be chosen.

Mobile Gray Box Security Assessment

This methodology aims at identifying vulnerabilities that can be exploited using applications on mobile phones. The assessments attempts at hacking into the application both as a registered user and an anonymous user. This also tests the application's resilience against reverse engineer attacks, and leverages both open source and commercial tools. Testers build custom threat profiles to discover contextual security vulnerabilities that are specific to the application.

Mobile White Box Security Assessment

Mobile White Box Security Assessment for IOS/Android aims at identifying vulnerabilities at the source code level. The assessments attempts at finding vulnerabilities from the coding or design flaws and the exploits the identified vulnerabilities as a registered user and an anonymous user.

This type of security assessment leverages automated scripts and tools to analyze source code. This type of assessment aims at identifying backdoor and suspicious code, weak algorithm and cryptographic usage. Testers build custom threat profiles to discover contextual security vulnerabilities that are specific to the application.

Deliverables

The Mobile Application vulnerability testing deliverables include:

- Detailed report on all performed testing activities
- The list of detected protection techniques and comments about their potential risks
- The list of detected cybersecurity problems with priorities and risks explanations
- The list of recommendations on how to resolve detected problems and improve solutions

Conclusion

Enterprises focus on developing mobile application to address their business needs, however in order to gain a competitive edge; security issues concerning mobile applications must be addressed. It is extremely important to examine these issues throughout development lifecycle, and ensure that any such risks are adequately mitigated. OWASP and other known security forums periodically release guidelines for securing mobile applications. All these guidelines should be diligently followed by developers and a structured mobile application security testing program should be implemented.

Let's discuss Securing Your IT and digital assets, Securing Your Brand.

Scoping Questionnaire for Penetration Testing

Penetration tests can range in a number of varieties from testing one application based on known vulnerabilities to far-reaching tests where no vulnerability information is provided and every system and network is in-scope. Additionally, a penetration can go as far as to gain control of the system by any means (aggressive) or to simply illustrate that it "could" be done by "taking these next steps", without actually taking the steps.

The following questions are intended to determine and refine the scope and extent of a desired penetration test. This template should be reviewed by our client and answered as thoroughly as possible. In the event that the client is not able to answer these questions, it is recommended that a **PLETHORA security** practitioner review each question with the client to ensure adequate information is obtained.

As per law, it is required that **PLETHORA team** obtain written permission by an authorized representative of the client to perform a penetration/security assessment. Please reference Appendix A entitled, Security Testing and Penetration Testing Authorization Agreement.

#	QUESTIONS	ANSWER	COMMENTS
1	What is the business requirement for this penetration test? 1. This is required by a regulatory audit or standard? 2. Proactive internal decision to determine all weaknesses? For example, is the driver for this to comply with an audit requirement, or are you seeking to proactively evaluate the security in your environment?		
2	Will this be a white box test or a black box test? **White Box** can be best described as a test where specific information has been provided in order to focus the effort. **Black Box** can be best described as a test where no information is provided by the client and the approach is left entirely to the penetration tester (analyst) to determine a means for exploitation.		
3	How many IP addresses and/or applications are included as in-scope for this testing? Please list them, including multiple sites, etc.		
4	What are the objectives? a) Map out vulnerabilities b) Demonstrate that the vulnerabilities exist c) Test the Incidence Response d) Actual exploitation of a vulnerability in a network, system, or application. Obtain privileged access, exploit buffer overflows, SQL injection attacks, etc. This level of test would carry out the exploitation of a weakness and		

	can impact system availability. e) All of the above		
5	What is the "target" of the Penetration test? Is it; a) An Application b) A Website c) A Network d) Application and Network e) Wireless f) Other, please explain		
6	What protocol should be followed for alerting on vulnerabilities found? a) Wait until the end of the testing to report all vulnerabilities b) Report vulnerabilities as we find them c) Daily report on the status of the testing d) Report only critical findings immediately		
7	Will this testing be done on a production environment? You need to understand that certain exploitation of vulnerabilities to determine and/or prove a weakness could crash your system or cause it to reboot. **SIMMSS** team is not liable for downtime caused by proving the system's weakness to attack.		
8	If production environments must not be affected, does a similar environment (development and/or test systems) exist that can be used to conduct the pen test?		
9	Are the business owners aware of this pen test? Are key stakeholders (business owners) aware that the nature of a pen test is to attack the system as a hacker (or hostile actor) would in order to learn and prove the system's weakness?		
10	At what time do you want these tests to be performed? a) During business hours b) After business hours c) Weekend hours d) During system maintenance window		
11	Who is the technical point of contact, assuming this is not a covert (black box) test of the incident response function? Name and Cellular phone number (available during this project) Alternate Name and Cellular phone number (available during this project)		
12	Additional Information?		

Appendix A – Security Testing and Penetration Testing Authorization Agreement

Security Testing and Penetration Testing Authorization Agreement

To authorize technical security assessment or penetration testing, please complete this form and send scanned copy to **PLETHORA** Team.

Contact and Scope Definitions

Client/Company Name: *(please print)* _____

Technical Contact Name:_____

Technical Contact Telephone:_____

Technical Contact E-mail: _____

IP Addresses / Ranges to be tested: (please identify internal or external addresses)

_____

_____

_____

_____