NAM A BANK

NAM A BANK – HỘI SỞ
201-203 Cách Mạng Tháng Tám, Quận 3, TP.HCM
[T] 028 3929 6699 - [F] 028 3929 66 88
[Hotline] 1900 6679 - [W] www.namabank.com.vn

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Số: 1868/2024/TB-NHNA-37

TP.HCM, ngày 13 tháng 06 năm 2024

THƯ MỜI CHÀO GIÁ

Đánh giá an toàn bảo mật hệ thống Internet Banking Nam A Bank

Kính gửi: QUÝ CÔNG TY

Ngân hàng TMCP Nam Á (Nam A Bank) có nhu cầu triển khai đánh giá an toàn bảo mật hệ thống Internet Banking, hình thức lựa chọn đối tác theo phương thức chào giá cạnh tranh.

Nay Nam A Bank kính mời các Công ty có đủ năng lực, kinh nghiệm và điều kiện tham gia chào giá cạnh tranh theo các nội dung hướng dẫn cụ thể như sau:

I. Mô tả yêu cầu

1. Nội dung công việc:

- Tấn công thử nghiệm (pentest) hệ thống Internet Banking của Nam A Bank, bao gồm nền tảng web (https://ops.namabank.com.vn) và mobile (Android và iOS).
- Tuân thủ các các tiêu chuẩn của OWASP về đánh giá bảo mật nền tảng web và mobile (Android và iOS).
  Tiêu chuẩn OWASP về đánh giá bảo mật nền tảng web và mobile (Android và iOS) được đính kèm thông báo này.

2. Tiêu chuẩn về đối tác:

- Đối tác phải là doanh nghiệp trong danh sách được cấp phép còn hiệu lực của Bộ TTTT (Danh sách doanh nghiệp đã được cấp giấy phép kinh doanh sản phẩm, dịch vụ an toàn thông tin mạng)

  Tham khảo: *https://mic.gov.vn/danh-sach-doanh-nghiep-da-duoc-cap-giay-phep-kinh-doanh-san-pham-dich-vu-an-toan-thong-tin-mang-197139244.htm*
- Có kinh nghiệm trên 5 năm trong lĩnh vực tấn công thử nghiệm.
- Đã từng thực hiện các dự án tương tự cho các ngân hàng trong vòng 1 năm tính từ thời điểm chào giá.

3. Thời gian thực hiện dự kiến:

- Thời gian dự kiến thực hiện từ tháng 07/2024.

4. Các điều kiện khác:

- Nam A Bank được quyền thay đổi các yêu cầu kĩ thuật và điều kiện triển khai theo nhu cầu thực tế để đảm bảo đem lại hiệu quả cao nhất cho hệ thống Internet Banking.

II. Một số yêu cầu về Hồ sơ chào giá (HSCG)

- Hồ sơ chào giá phải được đóng gói trong phong bì dán kín, các giấy tờ trong Hồ sơ phải có chữ ký của Người đại diện theo pháp luật và đóng dấu của Công ty. Hồ sơ chào giá bao gồm 01 bản gốc và 02 bản sao;
- Hồ sơ chào giá bao gồm:
  + Tài liệu minh chứng về tiêu chuẩn đã nêu tại mục I.2 *Tiêu chuẩn về đối tác*;
  + Bảng chào giá chi tiết.
- Đồng tiền trong hồ sơ chào giá: bằng đồng Việt Nam, giá chào phải bao gồm tất cả các khoản thuế, phí cần thiết;
- Hồ sơ chào giá phải có hiệu lực tối thiểu 60 ngày;

III. Thời gian và địa điểm nhận HSCG

- Thời gian gửi Hồ sơ chào giá: trước 17h00 giờ ngày 20/06/2024.
  Các Hồ sơ gửi sau thời gian nêu trên, Nam A Bank sẽ xem xét chấp nhận hoặc không chấp nhận tùy theo tình hình thực tế và phù hợp với quy định.
- Nơi nhận Hồ sơ: Phòng An ninh thông tin, Nam A Bank tại số 201-203 Cách Mạng Tháng Tám, P. 04, Q. 3, TP. Hồ Chí Minh.

Nam A Bank kính mời Quý Công ty quan tâm nộp Hồ sơ chào giá theo thời gian và địa chỉ nêu trên

Các vướng mắc trong quá trình lập Hồ sơ chào giá, Quý Công ty vui lòng liên hệ Ông Nguyễn Anh Dũng – Trưởng phòng An ninh thông tin (ĐT: 0827 827 777 – email: dungna@namabank.com)

PHÓ TỔNG GIÁM ĐỐC
KIÊM GIÁM ĐỐC KHỐI CNTT

Nguyễn Vĩnh Tuyên

PHỤ LỤC: TIÊU CHUẨN OWASP VỀ ĐÁNH GIÁ BẢO MẬT NỀN TẢNG WEB VÀ MOBILE (ANDROID VÀ IOS)

Mobile Application Security Requirements - iOS

ID	MSTG-ID	PRIORITY	DETAILED VERIFICATION REQUIREMENT
V1			ANTI-REVERSE ENGINEERING
1.1	MSTG-RESILIENCE-1	5	The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app.
1.2	MSTG-RESILIENCE-2	5	The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered.
1.3	MSTG-RESILIENCE-3	4	The app detects, and responds to, tampering with executable files and critical data within its own sandbox.
1.4	MSTG-RESILIENCE-4	1	The app detects, and responds to, the presence of widely used reverse engineering tools and frameworks on the device.
1.5	MSTG-RESILIENCE-5	5	The app detects, and responds to, being run in an emulator.
1.6	MSTG-RESILIENCE-6	4	The app detects, and responds to, tampering the code and data in its own memory space.
1.7	MSTG-RESILIENCE-7	0	The app implements multiple mechanisms in each defense category (8.1 to 8.6). Note that resiliency scales with the amount, diversity of the originality of the mechanisms used.
1.8	MSTG-RESILIENCE-8	0	The detection mechanisms trigger responses of different types, including delayed and stealthy responses.
1.9	MSTG-RESILIENCE-9	3	Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis.
1.10	MSTG-RESILIENCE-10	1	The app implements a 'device binding' functionality using a device fingerprint derived from multiple properties unique to the device.
1.11	MSTG-RESILIENCE-11	1	All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. Trivial static analysis does not reveal important code or data.
V2			ARCHITECTURE, DESIGN AND THREAD MODELLING
2.1	MSTG-ARCH-1	0	All app components are identified and known to be needed.
2.2	MSTG-ARCH-2	1	Security controls are never enforced only on the client side, but on the respective remote endpoints.
2.3	MSTG-ARCH-3	0	A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.
2.4	MSTG-ARCH-4	0	Data considered sensitive in the context of the mobile app is clearly identified.
2.5	MSTG-ARCH-5	0	All app components are defined in terms of the business functions and/or security functions they provide.
2.6	MSTG-ARCH-6	0	A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.
2.7	MSTG-ARCH-7	0	All security controls have a centralized implementation.
2.8	MSTG-ARCH-8	0	There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.
2.9	MSTG-ARCH-9	0	A mechanism for enforcing updates of the mobile app exists.

2.10	MSTG-ARCH-10	0	Security is addressed within all parts of the software development lifecycle.
2.11	MSTG-ARCH-11	0	A responsible disclosure policy is in place and effectively applied.
2.12	MSTG-ARCH-12	0	The app should comply with privacy laws and regulations.
2.13		2	A mechanism for enforcing updates of the mobile app exists.
V3			DATA STORAGE AND PRIVACY
3.1	MSTG-STORAGE-1	5	System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys.
3.2	MSTG-STORAGE-2	5	No sensitive data should be stored outside of the app container or system credential storage facilities.
3.3	MSTG-STORAGE-3	5	No sensitive data is written to application logs.
3.4	MSTG-STORAGE-4	2	No sensitive data is shared with third parties unless it is a necessary part of the architecture.
3.5	MSTG-STORAGE-5	2	The keyboard cache is disabled on text inputs that process sensitive data.
3.6	MSTG-STORAGE-6	3	No sensitive data is exposed via IPC mechanisms.
3.7	MSTG-STORAGE-7	5	No sensitive data, such as passwords or pins, is exposed through the user interface.
3.8	MSTG-STORAGE-8	5	No sensitive data is included in backups generated by the mobile operating system.
3.9	MSTG-STORAGE-9	2	The app removes sensitive data from views when moved to the background.
3.10	MSTG-STORAGE-10	2	The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.
3.11	MSTG-STORAGE-11	3	The app enforces a minimum device-access-security policy, such as requiring the user to set a device passcode.
3.12	MSTG-STORAGE-12	0	The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app.
3.13	MSTG-STORAGE-13	0	No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory.
3.14	MSTG-STORAGE-14	3	If sensitive data is still required to be stored locally, it should be encrypted using a key derived from hardware backed storage which requires authentication.
3.15	MSTG-STORAGE-15	0	The app's local storage should be wiped after an excessive number of failed authentication attempts.
V4			CRYPTOGRAPHY
4.1	MSTG-CRYPTO-1	5	The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.
4.2	MSTG-CRYPTO-2	2	The app uses proven implementations of cryptographic primitives.
4.3	MSTG-CRYPTO-3	0	The app uses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that adhere to industry best practices.
4.4	MSTG-CRYPTO-4	5	The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes.
4.5	MSTG-CRYPTO-5	0	The app doesn't re-use the same cryptographic key for multiple purposes.
4.6	MSTG-CRYPTO-6	5	All random values are generated using a sufficiently secure random number generator.
V5			AUTHENTICATION AND SESSION MANAGEMENT
5.1	MSTG-AUTH-1	5	If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint.

5.2	MSTG-AUTH-2	4	If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.
5.3	MSTG-AUTH-3	4	If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm.
5.4	MSTG-AUTH-4	4	The remote endpoint terminates the existing session when the user logs out.
5.5	MSTG-AUTH-5	3	A password policy exists and is enforced at the remote endpoint.
5.6	MSTG-AUTH-6	5	The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.
5.7	MSTG-AUTH-7	5	Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire.
5.8	MSTG-AUTH-8	2	Biometric authentication, if any, is not event-bound (i.e. using an API that simply returns "true" or "false"). Instead, it is based on unlocking the keychain/keystore.
5.9	MSTG-AUTH-9	5	A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.
5.10	MSTG-AUTH-10	5	Sensitive transactions require step-up authentication.
5.11	MSTG-AUTH-11	3	The app informs the user of all sensitive activities with their account. Users are able to view a list of devices, view contextual information (IP address, location, etc.), and to block specific devices.
5.12	MSTG-AUTH-12	4	Authorization models should be defined and enforced at the remote endpoint.
V6			**NETWORK COMMUNICATION**
6.1	MSTG-NETWORK-1	5	Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.
6.2	MSTG-NETWORK-2	0	The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.
6.3	MSTG-NETWORK-3	5	The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.
6.4	MSTG-NETWORK-4	5	The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.
6.5	MSTG-NETWORK-5	2	The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.
6.6	MSTG-NETWORK-6	1	The app only depends on up-to-date connectivity and security libraries.
V7			**PLATFORM INTERACTION**
7.1	MSTG-PLATFORM-1	4	The app only requests the minimum set of permissions necessary.
7.2	MSTG-PLATFORM-2	5	All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.
7.3	MSTG-PLATFORM-3	5	The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected.
7.4	MSTG-PLATFORM-4	5	The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected.
7.5	MSTG-PLATFORM-5	4	JavaScript is disabled in WebViews unless explicitly required.
7.6	MSTG-PLATFORM-6	3	WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.
7.7	MSTG-PLATFORM-7	4	If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.

7.8	MSTG-PLATFORM-8	2	Object deserialization, if any, is implemented using safe serialization APIs.
7.9	MSTG-PLATFORM-9	3	The app protects itself against screen overlay attacks. (Android only)
7.10	MSTG-PLATFORM-10	0	A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed.
7.11	MSTG-PLATFORM-11	0	Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered.
V8			CODE QUALITY AND BUILD SETTINGS
8.1	MSTG-CODE-1	5	The app is signed and provisioned with a valid certificate, of which the private key is properly protected.
8.2	MSTG-CODE-2	5	The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).
8.3	MSTG-CODE-3	2	Debugging symbols have been removed from native binaries.
8.4	MSTG-CODE-4	3	Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages.
8.5	MSTG-CODE-5	2	All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities.
8.6	MSTG-CODE-6	2	The app catches and handles possible exceptions.
8.7	MSTG-CODE-7	2	Error handling logic in security controls denies access by default.
8.8	MSTG-CODE-8	1	In unmanaged code, memory is allocated, freed and used securely.
8.9	MSTG-CODE-9	1	Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.
V9			FUNCTIONALITY
9.1		5	Poor or missing authorization between privileges or roles
9.2		5	In React Native app, source code index.main.bundle is not compiled by using hermes tool.
9.3		5	Extraneous functionality in mobile app lead to hidden functionality in in backend systems

OWASP: Testing Guide v4.2 Checklist

Information Gathering	Test Name
WSTG-INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage
WSTG-INFO-02	Fingerprint Web Server
WSTG-INFO-03	Review Webserver Metafiles for Information Leakage
WSTG-INFO-04	Review Webserver Metafiles for Information Leakage
WSTG-INFO-05	Review Webpage Content for Information Leakage

WSTG-INFO-06	Identify application entry points
WSTG-INFO-07	Map execution paths through application
WSTG-INFO-08	Fingerprint Web Application Framework
WSTG-INFO-09	Fingerprint Web Application
WSTG-INFO-10	Map Application Architecture
Configuration and Deploy Management Testing	
WSTG-CONF-01	Test Network Infrastructure Configuration
WSTG-CONF-02	Test Application Platform Configuration
WSTG-CONF-03	Test File Extensions Handling for Sensitive Information
WSTG-CONF-04	Review Old Backup and Unreferenced Files for Sensitive Information
WSTG-CONF-05	Enumerate Infrastructure and Application Admin Interfaces
WSTG-CONF-06	Test HTTP Methods
WSTG-CONF-07	Test HTTP Strict Transport Security
WSTG-CONF-08	Test RIA cross domain policy
WSTG-CONF-09	Test File Permission
WSTG-CONF-10	Test for Subdomain Takeover
WSTG-CONF-11	Test Cloud Storage
Identity Management Testing	
WSTG-IDNT-01	Test Role Definitions
WSTG-IDNT-02	Test User Registration Process
WSTG-IDNT-03	Test Account Provisioning Process
WSTG-IDNT-04	Testing for Account Enumeration and Guessable User Account
WSTG-IDNT-05	Testing for Weak or unenforced username policy
Authentication Testing	
WSTG-ATHN-01	Testing for Credentials Transported over an Encrypted Channel
WSTG-ATHN-02	Testing for Default Credentials
WSTG-ATHN-03	Testing for Weak Lock Out Mechanism
WSTG-ATHN-04	Testing for Bypassing Authentication Schema
WSTG-ATHN-05	Testing for Vulnerable Remember Password

WSTG-ATHN-06	Testing for Browser Cache Weaknesses
WSTG-ATHN-07	Testing for Weak Password Policy
WSTG-ATHN-08	Testing for Weak Security Question Answer
WSTG-ATHN-09	Testing for Weak Password Change or Reset Functionalities
WSTG-ATHN-10	Testing for Weaker Authentication in Alternative Channel
Authorization Testing	
WSTG-ATHZ-01	Testing Directory Traversal File Include
WSTG-ATHZ-02	Testing for Bypassing Authorization Schema
WSTG-ATHZ-03	Testing for Privilege Escalation
WSTG-ATHZ-04	Testing for Insecure Direct Object References
Session Management Testing	
WSTG-SESS-01	Testing for Session Management Schema
WSTG-SESS-02	Testing for Cookies Attributes
WSTG-SESS-03	Testing for Session Fixation
WSTG-SESS-04	Testing for Exposed Session Variables
WSTG-SESS-05	Testing for Cross Site Request Forgery
WSTG-SESS-06	Testing for Logout Functionality
WSTG-SESS-07	Testing Session Timeout
WSTG-SESS-08	Testing for Session Puzzling
WSTG-SESS-09	Testing for Session Hijacking
Data Validation Testing	
WSTG-INPV-01	Testing for Reflected Cross Site Scripting
WSTG-INPV-02	Testing for Stored Cross Site Scripting
WSTG-INPV-03	Testing for HTTP Verb Tampering
WSTG-INPV-04	Testing for HTTP Parameter Pollution
WSTG-INPV-05	Testing for SQL Injection
WSTG-INPV-06	Testing for LDAP Injection
WSTG-INPV-07	Testing for XML Injection
WSTG-INPV-08	Testing for SSI Injection
WSTG-INPV-09	Testing for XPath Injection