Service Definition
Application Security Testing

Introduction

NTA's application security testing service aims to ensure that the application (e.g. public-facing transactional website, mobile application, internal CMS, cloud based solution) is securely configured, thus preventing an attacker from gaining access or a user being exposed to confidential or sensitive data, another user's account or the back end database.

This service allows organisations who have procured services via the Digital Marketplace, or who are using cloud hosted software, infrastructure or platforms, to gain independent analysis and information security assurance regarding the governance and controls that are in place to protect these services and systems. Such assurance is vital for cloud based services which possess specific security considerations due to their on-demand, remotely accessible and multi-tenanted attributes.

This specialist testing service provides customers with governance support in relation to the integration of cloud based services and systems into their organisations.

Scoping

A scoping document will be provided prior to purchase of the service in order to fully understand the functionality, configuration and size of the application and possible attack/threat vectors. A recommended number of units/days will then be provided to enable requirements to be met.

Service Overview

Testing will begin with network level testing of the hosted web server(s) to check for common vulnerabilities (e.g. vulnerable software versions). Unauthenticated application testing will then be performed from the perspective of an external attacker with no privileged information, with particular attention paid to the authentication/protection mechanisms deployed.

Post-authenticated testing will be conducted using valid credentials for each user type to ensure user permissions are enforced, i.e. determine if we can gain access to other user accounts within the system and users are not able to access information or areas they are not granted permissions to see/access.

Testing will take the approach of simulating possible attack scenarios, such as attempts to gain unauthorised access to information through privilege escalation or weak protocols. We will also look at the controls around administrative actions and how they are managed, which will include an assessment of patch levels, weak passwords and services offered, as well as passwords, the least privilege principle (e.g. over-privileged accounts, including service accounts), ports, protocols and services – and access controls, autologon, weak ciphers, SNMP default community names etc.

Testing will incorporate an assessment, where appropriate (and except where testing would contravene Microsoft Cloud Penetration Testing Rules of Engagement), against the Open Web Application Security Project (OWASP) Top 10 – the ten most critical web application security risks – which as of the latest 2021 report are as follows:

A01:2021 – Broken Access Control
A02:2021 – Cryptographic Failures

Service Definition
Application Security Testing

A03:2021 – Injection
A04:2021 – Insecure Design
A05:2021 – Security Misconfiguration
A06:2021 – Vulnerable and Outdated Components
A07:2021 – Identification and Authentication Failures
A08:2021 – Software and Data Integrity Failures
A09:2021 – Security Logging and Monitoring Failures
A10:2021 – Server-Side Request Forgery (SSRF)

The test consultant will move through the following methodology as appropriate:

Initial Reconnaissance
Applications can unintentionally leak information about their configuration and internal workings. Often, this information can be leveraged to launch or even automate more powerful attacks. NTA will determine the components of the application and then examine their configuration in detail. This will include fingerprinting of the web server, reviewing metafiles and information it provides, enumerate applications on the web server, identify entry points on the web application, fingerprint web application framework and mapping the application architecture.

Web Server Vulnerabilities
Web application security can be compromised due to misconfigurations on the underlying web server. These vulnerabilities may vary between an unauthenticated and authenticated session, so testing will consider both perspectives in order to determine if there are vulnerabilities that a malicious user could exploit.

Encryption
Encryption must be used for all secure data transmissions. NTA will determine the combinations of protocols, ciphers and hashes the application's web server supports and perform checks to discover if it can be made to negotiate a weakly encrypted or unencrypted channel that may put sensitive data at risk. We will identify the key places on the application where data is transmitted (e.g. during authentication, when submitting data, etc.) and verify that this transmission is being correctly encrypted.

Account Sign-up & Account Management
If the application allows users to sign-up for an account online, we will assess the security of the sign-up process from both the user and application perspectives. We will check whether attackers can use this feature maliciously to disrupt the normal functioning of the application, or whether users can put themselves at risk through choosing weak usernames and passwords.

If the application also provides account management facilities, we will assess how secure these are and whether users could put themselves at risk by using them unwisely. We will check there is adequate user guidance on how to use these facilities securely, and whether it is possible to choose weak passwords, or leave accounts open to compromise in some other way.

Service Definition
Application Security Testing

Authentication and Session Management
Flaws in this area most frequently involve the failure to protect credentials and session tokens through their lifecycle. These flaws can lead to the hijacking of user or administrative accounts, undermine authorisation and accountability controls and cause privacy violations.

Flaws in the main authentication mechanism are typically introduced through ancillary authentication functions, such as logout, password management, timeout, remember me, secret question and account updates. The following different vulnerabilities will be investigated:

- Bypassing the authentication mechanism
- Cookie hijacking
- Session fixation
- Exposed session variables

We will assess whether accounts are locked out if too many incorrect attempts are made to log in and check whether mechanisms are in place to protect users from covert attacks on their account such as shoulder surfing or key-stroke logging.

Authorisation Testing
If there are user types with different privilege levels, authorisation testing will confirm that the appropriate access to resources is only permitted to the intended user. NTA will assess the authorisation process and using the information gathered, try to circumvent the authorisation mechanism(s). The following will be investigated:

- Directory traversal
- Privilege escalation
- Insecure direct object references (*further details below*)

Cross Site Request Forgery (CSRF)
This attack forces an end user to execute unwanted actions on a web application in which they are currently authenticated. Checks will made to ensure end-user data and system operation cannot be compromised.

Data Sanitisation and Error Checking (SQL injections, XSS etc)
Tests will be performed to see how the application copes with unexpected data. Sometimes servers will return useful information in error messages, or can be made to return output, which could cause arbitrary code to be run on either the server or the client. This could happen accidentally if a user edits their internal URL, or maliciously if a user wants to access data that does not belong to them. Below are examples of the tests that will be performed.

- ***Cross-site scripting (XSS)***
  XSS errors may allow an attacker to execute scripts in the browser, which can hijack user sessions, deface web sites, insert hostile content, conduct phishing attacks and take over the user's browser with scripting malware. The three types of XSS vulnerabilities are covered.

- ***Injection flaws***

  NTA will examine the application for the various types, such as: SQL, LDAP, XPath, XSLT, HTML, XML, OS command injection, SOAP and SSI.

- ***URL Access Restriction***

  Exposed page links must not be presented to unauthorised users. NTA will look for access control checks are being performed correctly, before a request to a sensitive function is granted. This method of "forced browsing" encompasses guessing links and brute force techniques to find unprotected pages.

- ***Improper Data Validation***

  The most common web application security weakness is the failure to properly validate input. This weakness leads to almost all of the major vulnerabilities in web applications, such as cross site scripting, SQL injection, interpreter injection, locale/Unicode attacks, file system attacks and buffer overflows.

- ***Insecure Direct Object Reference***

  NTA checks that an unauthorised user cannot manipulate direct object references to access or traverse other objects without authorisation within a file, directory, database record, or key, as a URL or form parameter. It is found that many applications expose their internal object references to users. Attackers use parameter tampering to change references and violate the intended but unenforced access control policy. Frequently, these references point to file systems and databases, but any exposed application construct could be vulnerable.

- ***Malicious file execution***

  It is found that, on many platforms, frameworks allow the use of external object references, such as URLs or file system references. When the data is insufficiently checked, this can lead to arbitrary remote and hostile content being included, processed or invoked by the web server.

  If there is a file upload facility within the application NTA will use a test file called EICAR, which mimics the characteristics of a malicious file (malware) but without any harmful content to effectively test the performance of the antivirus solution in place.

API & Web Services

A vulnerability analysis will be performed to confirm secure configuration of the APIs/web services in scope. Testing will consider, but not be limited to:

- General security concerns via the APIs
- Can access to the APIs be achieved without authorisation (e.g. a valid token)?
- Are internal/backend services accessible from outside sources?
- Can damaging information be passed through the APIs to threaten backend services?

As well as ensuring data is being exchanged with appropriate encryption in place, the following tests will be performed:

- Acquire valid data and perform valid requests to gather possible information leakage regarding back end systems, aiding further tests.
- Perform validation tests on interface arguments

Although response parameters via APIs will tend to be structured data, such as XML or JSON, the request-and-response process still runs over HTTP, and the structured data may still be displayed as HTML. Vulnerabilities are typically sought out by firstly determining and analyzing the available requests and thus the potential attack surface, and then by sending unintended data through the identified parameters and observing the response this generates.

Therefore, the vulnerabilities NTA will seek to identify within the APIs will focus on injection, session management and parameter tampering issues as previously described in the 'Authentication and Session Management' and 'Data Sanitisation and Error Checking' sections.

Deliverables
The output is a tailor-written formal report containing:

- An executive overview with business implications
- A summary of risks identified, ordered from high to low severity
- Technical details of each issue found
- Recommendations for closing holes found
- Screen shots and supporting evidence for risks found

The style and format of the report is designed with clarity and ease of use in mind. Features include hyperlinks to relevant sections of the document as well as external sources where applicable.

Additional support is provided in the form of an optional telephone technical debrief plus free of charge retesting of any high risk issues identified.

Service Management
You will be appointed a dedicated account manager and provided with points of contact for commercial and technical questions, to ensure that the service is properly executed.

NTA has a Service Level Agreement that is monitored by a R.A.G system to ensure that delivery standards are met on all projects. The SLA includes timeframes for project delivery with technical staff working to targets to ensure that deadlines are met.

Additionally, NTA has a customer services department that is responsible for the key administrative aspects of delivery such as liaising with you to confirm site details, scheduling the test, ensuring reports are sent out in a timely manner, scheduling follow up tests and debriefs etc. This is documented and monitored through the use of an internal database system, and links into other key processes regarding the correct method of sending the report according to the protective marking of the contents.

Service Definition
Application Security Testing

Ordering and Invoicing Process
Once the scope of work and number of units/days required has been agreed between the customer and NTA, a valid purchase order is required to enable scheduling and delivery.

An invoice will be raised upon delivery of the service.

Customer Responsibilities
The customer is responsible for the completion of a Technical Details Form (TDF) prior to delivery. The provision of appropriate information such as target URLs and valid credentials for each of the user roles to be evaluated will ensure that delivery meets expectations and requirements.

Test Prerequisites
- Provide purchase order/written acceptance of this quotation.
- Confirm hoster permission for testing has been obtained (if applicable/required).
- Complete a Technical Details Form with contact details, target URL(s), user credentials for test accounts as appropriate (n/a if users can sign-up/register for an account unless Admin user is to be tested), API endpoint call examples, relevant documentation (e.g. Swagger, WSDL, Postman Project, etc.) API authentication token(s)/key(s); at least five working days in advance of the test start date.
- The target application(s) should be fully functional and populated with data (test or live) at the time of the test. This is to ensure the application(s) can be used as intended e.g. a search yields a corresponding result.
- The application(s) should also be stable and free from upgrades and configuration changes, although it is fine for the application(s) to be used normally during the test window, either as a live application or in UAT.
- Where multiple non-intuitive user journeys exist, workflow documentation should be provided.
- Sample input data should also be provided where appropriate i.e. data that will be accepted in initial forms, such as a reference or account number, in order that the tester may sequence through to subsequent pages.
- If user groups have different access rights, a permissions matrix is useful in order to support tests that attempt to break out of defined privilege levels.
- Please note **that tests will submit data and create logs** and that, where Captchas or form controls are not used, this may result in a high volume of submissions and record creation in back end systems. For authenticated testing, for which clients provide NTA with user accounts, please ensure these are test accounts that can have data deleted or reversed from linked systems following the test. For pre-authenticated testing, please ensure public forms are protected from bulk-registration or inform NTA prior to the test of the need for caution.
- Allow Intertek NTA's source IPs if necessary to externally access application across the Internet and through standard HTTPS.