SPECIFICAȚII TEHNICE

Anexa nr. 22
la Documentația standard nr. 115 din 15.09.2021

Numărul procedurii de achiziție: ***Licitație deschisă 21469825/ ocds-b3wdp1-MD-1755885385826***

Denumirea licitaţiei: ***Servicii de testare de securitate pentru sisteme informaționale***

Nr. d/o	Denumirea bunurilorşi/sau a serviciilor	Modelul articolului	Ţara de origine	Produ-cătorul	Specificarea tehnică deplină solicitată de către autoritatea contractantă	Specificarea tehnică deplină propusă de către ofertant	Standarde de referință
	1	2	3	4	5	6	7
	Bunuri/Servicii: ***Servicii de testare de securitate pentru sisteme informaționale***						
1	**Servicii de testare de securitate pentru sisteme informaționale**				Conform caietului de sarcini (C.1)	În conformitate cu cerințele expuse în Anunț și Caiet de sarcini și Descrierii tehnice anexate.	ISO 9001:2015 ISO 27001

Semnat:________________
Numele, Prenumele: Ghincu Sergiu
În calitate de: Director General
Ofertantul: DAAC System Integrator S.R.L.
Adresa: mun.Chisinau str.Calea Iesilor 10

SOCIETATEA COMERCIALĂ * REPUBLICA MOLDOVA
"DAAC SYSTEM INTEGRATOR" S.R.L.
IDNO 1006600054871

Digitally signed by Culicov Iuri
Date: 2025.09.19 16:46:59 EEST
Reason: MoldSign Signature
Location: Moldova
MOLDOVA EUROPEANĂ

Cyber-Security Assessments Services Proposal

for (Agenția de Guvernare Electronică (AGE)

-TECHNICAL PART-

Cover Letter

Dear Sir/Madam,

We are pleased to submit to you a comprehensive commercial proposal for our services in relation to your request.

We understand that the objective of this procurement is to engage a service provider for comprehensive security assessment, including vulnerability testing, penetration testing, and security audits, covering AGE's systems, platforms, infrastructure, and application modules. The goal is to ensure and validate a high level of security, confidentiality, integrity, and availability for AGE's electronic public services. As such, we propose the use of experienced individuals who will ensure that the scope of the requested work is delivered as per the outlined and agreed framework.

We believe that the below proposal will meet all of your requirements. Given the opportunity, our team of experienced professionals will provide the required services to Agenția de Guvernare Electronică (AGE) and will ensure that sufficient knowledge transfer is provided to the Agenția de Guvernare Electronică (AGE) team. We are confident that we can meet expectations and continue to build a successful partnership in cyber security.

If you have any questions, please do not hesitate to contact me directly at +40752-607.204.

Sincerely,

Pusoiu Andrei

Director, DAAC Digital

Contents

Executive Summary

DAAC Digital understands Agenția de Guvernare Electronică (AGE)'s objective to engage a trusted service provider for comprehensive security assessments, including vulnerability testing, penetration testing, and security audits, across AGE's systems, platforms, infrastructure, and application modules. Our goal is to help AGE ensure and validate the highest levels of security, confidentiality, integrity, and availability for its electronic public services. To achieve this, we will leverage our expertise in Web and Mobile Security Assessments, as well as Security Configuration Reviews, applying internationally recognized methodologies and best practices. We commit to assigning experienced professionals who will deliver the full scope of services in line with the agreed framework and the highest international standards.

This proposal outlines our technical approach and compliance with the Annexes from RFQ.

DAAC Digital brings the extensive and relevant experience that is being sought to provide qualified IT security assessments. We have significant experience in the region as well as Europe and Asia, with companies including BNP Paribas (FR), Euronext (EU), Saltege (BG), Premialab (FR), ENBD Bank (UAE), ADIB (UAE), Du (UAE), Telecommunications Regulatory Authority (UAE), incl. Mashreq Bank. We have delivered high-quality services across the IT security portfolio to these entities. We also bring the lessons learned from other high-risk, continuous service industries that we have served, such as oil and gas and aviation. These projects have allowed us to refine both our technical offering and project management skillset. This enables DAAC Digital to continually improve deliveries, build a consistent track record and amass working knowledge of operations in the region.

We are experienced in dealing with the typical challenges, risks and issues that arise when working with customers like Agenția de Guvernare Electronică (AGE). This familiarity with the operating environment enables us to quickly address risks and elicit specific issues, as well as seek solutions to issues as they arise (and draw on past experiences). Our management and leadership are committed to creating a lasting partnership with Agenția de Guvernare Electronică (AGE) and we have selected project managers that we are confident can deliver service quality that Agenția de Guvernare Electronică (AGE) can trust. Finally, we believe our team members are our greatest asset and we have developed a team that is solution-focused, technically strong and experienced with working on some of the most complex security projects in the region.

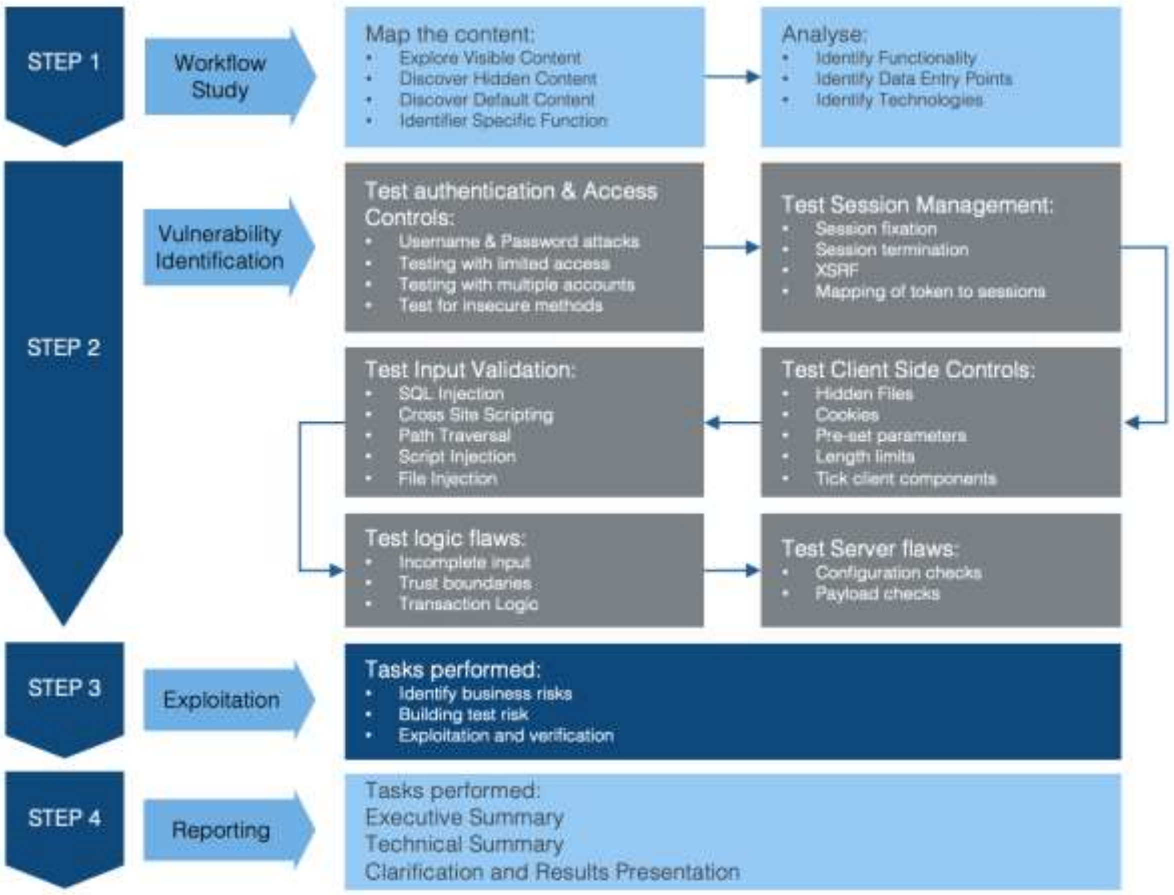
Based on recent discussions, DAAC Digital understands that Agenția de Guvernare Electronică (AGE) is seeking an experienced Information Security Consulting vendor with the necessary expertise and relevant experience in delivering multiple Security Services. DAAC Digital is pleased to submit the proposal for the requested Security Services as described in this document. DAAC Digital is submitting this proposal which includes a detailed approach and scope of work and in later stage this scope and commercials may change as per requirements.

Project Management and Services Methodology

This chapter presents DAAC Digital's different methodologies for different cyber-security services. The final project execution approach might be adjusted depending on actual scope, dependencies, and specific requirements or constraints given by Agenția de Guvernare Electronică (AGE) during the kick-off of a particular engagement.

General Approach

In order to meet all requirements requested by Agenția de Guvernare Electronică (AGE) every activity we divide into the following main steps:



The steps are aligned to the in-depth security concepts and are focused on process and technical security controls and their implementation in the various phases of the project delivery. The results provided for each activity will include detailed and comprehensive assessments of Agenția de Guvernare Electronică (AGE)'s security posture, expansive recommendations, and tools and knowledge to facilitate continuous improvements. For each activity, Agenția de Guvernare Electronică (AGE) will nominate relevant employees to ensure smooth knowledge transfer and prerequisites preparation.

The project phases and activities performed will also be aligned with international best practices and standards such as:

- Open Source Security Testing Methodology Manual (OSSTMM)
- Penetration Testing Execution Standard (PTES)
- Open Web Application Security Project (OWASP) Testing Guide
- The National Institute of Standards and Technology (NIST)
- PCI Data Security Standard Penetration Testing Guidance (PCI DSS)
- The Intelligence Lifecycle & F3EAD Cycle (Threat Intelligence)
- OWASP Mobile Security Testing Guide (MSTG)
- Penetration Testing Framework for IoT (PTFIoT)
- PCI DSS ATM Security Guidelines
- CIS Cloud Foundations Benchmark Standard
- OWASP Code Review Guide
- Threat Intelligence Based Ethical Red Teaming Framework (TIBER-EU)
- Application Security and Development Security Technical Implementation Guide
- Social Engineering Attack Framework and Toolkit (SET)
- Digital Forensics Framework (DFF)
- Incident Response Framework (NIST)
- Secure Controls Framework (SCF)
- CREST Penetration Testing Guide
- CSA STAR Self-Assessment / CAIQ
- CIS Secure Platforms Benchmarks (CIS Security)
- Application Security Verification Standard (ASVS)

Vulnerability Assessment/Penetration Test

Our vulnerability assessment/penetration test (VA/PT) service is designed to provide a comprehensive overview of technical security issues throughout your environment. Our services are aligned to the requirements set forth in standards and initiatives such as ISO/IEC 27001, COBIT 5, the ISF Standard of Good Practice, Open-Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP), National Institute of Standards and Technology Special Publication 800-115, the Information Systems Security Assessment Framework (ISSAF), and Penetration Testing Execution Standard (PTES). We can also provide a comparative gap analysis against relevant standards such as PCI/DSS, NERC, HIPAA, or other regulatory requirements as may be specified in the project scope.

The assessment will be conducted from an external perspective. Using industry-standard scanning tools as well as manual discovery techniques, our team will interact with and assess the security of each device in your network, in accordance with the scope of work. Each device will be assessed to determine the services that it offers, the versions of the associated software and hardware, the security configuration of the device, and the resulting security posture of the device.

Examples of the types of testing that may be performed during this phase include but are certainly not limited to:

- Network mapping and enumeration
- Port scanning
- Vulnerability scanning
- Local area network manipulation attempts
- IP address spoofing
- Vulnerability validation
- Authentication attacks
- Network monitoring

After initial vulnerability scans and associated tests are conducted, our engineers will analyze the results and perform additional manual testing. These additional tests are designed to verify initial findings, understand the potential business impact of the technical findings, and explore ways in which an attacker could potentially exploit the identified vulnerabilities, either directly or by combining a series of exploits into an attack chain, which can result in further penetration into the network and risk the business.

Security Configuration Review

While the technology deployed within your organization certainly contributes to your overall security posture, in many cases, the configuration of existing technology greatly enhances or destroys the security of your environment. Our security engineers can perform a comprehensive configuration review of your existing technologies (Electron, Erlang) including the databases, servers, and web servers, identifying opportunities to enhance security, functionality, and interoperability of your systems and improving your return on existing investment.

As determined by the scope of work, our team can examine device configurations across the network appliances, endpoints, and other devices including but not limited to:

- Firewalls
- Intrusion Detection/Prevention Systems
- Web proxies
- WebServers
- Web application
- Databases
- Servers

In addition to assessing and making recommendations for optimizing configurations to improve general security and functionality, our team can optionally perform a gap analysis of your system configurations against industry-standard hardening guidelines such as the Center for Internet Security benchmarks, the National Institute of Standards and Technology checklist program, or the Australian Cyber Security Centre hardening guidelines.

Web Application Penetration Testing

During this phase, a security expert from DAAC Digital will aim to identify additional potential weaknesses in the identified web/stand-alone applications. Based on the review, our experts will provide recommendations where required to allow our customers to quickly and cost-effectively raise the level of security. DAAC Digital has built a large part of its team that focuses solely on the dangers of Web applications and their interconnectivity. In order to usefully test the application from a crystal box point of attack, DAAC Digital will require minimum of two valid user names and passwords on any dynamic web applications to allow for thorough testing from every aspect.

The general methodology is outlined below. Some customizations of the methodology may be done, based on the type of application being tested.

The early stage of that phase includes the understanding of the business logic of the application. After the business logic has been determined, various tests will be performed to determine the 'relationship' between the different scripts that make up the application and identify possible ways to manipulate the logic of the application, e.g. by bypassing certain scripts that would do necessary security checks. The security tests to be performed will be based on the OWASP Application Security Verification Standard, which includes but is not limited to the following areas of testing.

Injection

During this phase, DAAC Digital experts will perform various tests to identify whether there are any injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Broken authentication

During this phase, the application functions related to authentication. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

XML External Entities (XXE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Security Misconfiguration

During this phase, DAAC Digital experts will review the secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained, as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application.

Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Insecure Deserialization

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

Insufficient Logging & Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

DAAC Digital Security's ASA includes, but is not limited to, the identification of the following risks:

Application Profiling and Information Disclosure	Platform and Third-Party Misconfiguration	Cookie and Session Handling
▪ Default Banners ▪ Unhandled Error Conditions ▪ Application Binary Information Leakage ▪ Extraneous Content in Web Backend ▪ Source Code Disclosure ▪ Unintended Data Leakage ▪ Content Expiration and Cache Control ▪ Insecure Data Storage ▪ Account Enumeration ▪ Backup/Archive Content	▪ Default Administrative Credentials ▪ Default Content and Scripts ▪ Web Server Vulnerabilities ▪ Weak SSL Implementation ▪ Flawed Use of Cryptography	▪ Session Fixation/Hijacking ▪ Set-Cookie Weaknesses ▪ Sensitive Information Disclosure ▪ Cookie Poisoning ▪ Multiple Simultaneous Login Allowed ▪ Session Timeout ▪ Explicit/Implicit Logout Failures ▪ Persistent Sessions ▪ Custom Session Management
Command Injection Flaws	**Logic Flaws**	**Client-Side Flaws**
▪ SQL Injection ▪ XXE, XPath, and XML Injection ▪ OS Command Injection ▪ Server Script Injection/Upload ▪ Cross-Site Scripting (XSS) ▪ Buffer Overflow	▪ Privilege Escalation ▪ Sensitive Information Disclosure ▪ Data Mining/Inference ▪ Functional Bugs ▪ Application-Specific Control Failures ▪ Weak Data Validation ▪ Race Conditions	▪ Exposure of Sensitive Business Logic ▪ Reliance on Client-Side Validation ▪ AJAX/Web Service Flaws ▪ Interprocess Communication Weaknesses ▪ Client-side Injection
Authentication and Authorization		

Application Profiling and Information Disclosure	Platform and Third-Party Misconfiguration	Cookie and Session Handling
▪ Unauthenticated Sensitive Content ▪ Poor Separation of Privilege ▪ Brute-Force Login ▪ Weak Password, Passcode, or PIN Policy ▪ Account Lockout/Denial of Service ▪ SSO Weaknesses ▪ Security Question Weaknesses ▪ Client-side Credential Storage ▪ Anti-automation Flaws		

DAAC Digital performs extensive manual testing, which comprises a significant majority of the testing effort. During this portion of the testing, the DAAC Digital consultant executes the application, and analyzes the communication, functions, and the data the application sends and receives. The DAAC Digital

Team tests complex interactions, workflows, and business logic. Additionally, the Team manually evaluates areas of the application and specific vulnerabilities that automated tools either have difficulty with or are unable to identify.

Mobile Application Security Assessment

A Mobile Application Security Assessment (MASA) is an assessment of the application on the running device as well as the associated back-end components. The objective of the assessment is to identify vulnerabilities within these environments and verify their existence using manual testing techniques. These assessments are most successful when customers share all known information with the tester; however, the customer can elect to share less information.

The MASA is a comprehensive assessment that identifies vulnerabilities ranging from Critical to Informational severity. DAAC Digital's Application Security Team identifies, verifies, and reports anything that raises the attack surface of the application. We will use multiple techniques to simulate attacks from both an unauthenticated and authenticated attacker perspective, exposing the greatest amount of attack surface and providing the most value from the testing efforts.

DAAC Digital follows a highly-structured methodology to ensure a thorough test of the application and its environment. Our methodology uses a phased approach consisting of information gathering, testing, verification, and notification. DAAC Digital Security employs a comprehensive and careful methodology in order to identify any potentially dangerous functionality. Prior to performing tests against these functions, DAAC Digital shares any potential impacts with the client. These steps ensure the least amount of business impact possible.

In cases where automated testing cannot be performed safely, multiple manual testing techniques may be used to confirm the existence of a vulnerability. If a deep level of exploitation of a database, application server, or host platform weakness is necessary to gather evidence of a vulnerability, DAAC Digital's tester will contact the engagement POC, as well as any applicable customer personnel. The DAAC Digital Team will discuss a plan of attack and any potential concerns raised, and will seek explicit approval from the client to exploit any vulnerability that has a potential to impact production operations.

The mobile application testing methodology is based from the OWASP mobile security project and covers all aspects of the OWASP Mobile Top 10 for 2017 and incorporates experience and testing techniques used in other areas of security testing. Besides this, as a walkthrough check-list that the security team is using the Mobile Security Testing Guide (MSTG) which is a comprehensive manual for mobile app security development, testing and reverse engineer.

An overview of the mobile security testing methodology is documented below, as in the OWASP TOP 10:

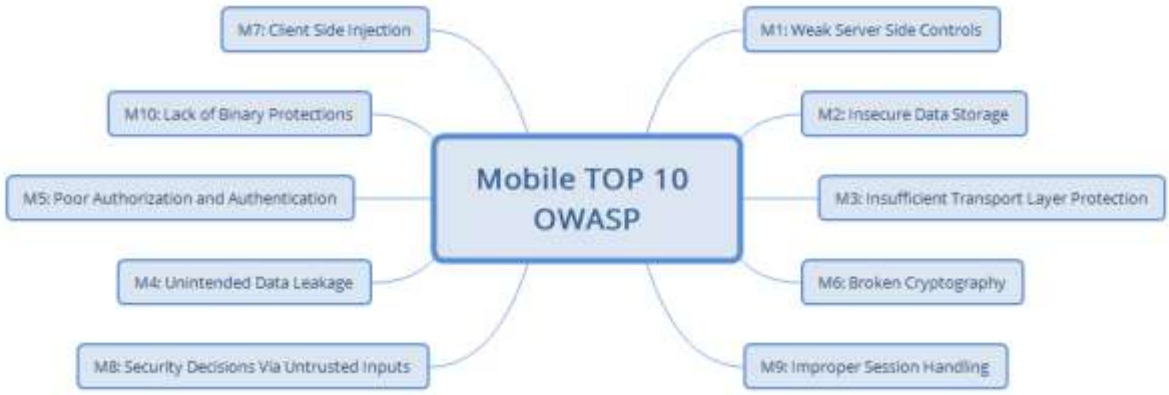

M7: Client Side Injection
M10: Lack of Binary Protections
M5: Poor Authorization and Authentication
M4: Unintended Data Leakage
M8: Security Decisions Via Untrusted Inputs
Mobile TOP 10 OWASP
M1: Weak Server Side Controls
M2: Insecure Data Storage
M3: Insufficient Transport Layer Protection
M6: Broken Cryptography
M9: Improper Session Handling

Figure 1 - OWASP Top 10 for Mobile Security Assessment

DAAC Digital Approaches to Mobile App Assessments

We strongly advise that the client shares the source code so that our team can use the testing time as efficiently as possible. The tester's code access obviously doesn't simulate an external attack, but it simplifies the identification of vulnerabilities by allowing the tester to verify every identified anomaly or suspicious behaviour at the code level. A white-box test is the way to go if the app hasn't been tested before. However, there are other approaches that ca be used:

Black-box testing is conducted without the tester's having any information about the app being tested. This process is sometimes called "zero-knowledge testing". The main purpose of this test is allowing the tester to behave like a real attacker in the sense of exploring possible uses for publicly available and discoverable information.
White-box testing (sometimes called "full knowledge testing") is the total opposite of black-box testing in the sense that the tester has full knowledge of the app. The knowledge may encompass source code, documentation, and diagrams. This approach allows much faster testing than black-box testing due to its transparency and with the additional knowledge gained a tester can build much more sophisticated and granular test cases.
Gray-box testing is all testing that falls in between the two aforementioned testing types: some information is provided to the tester (usually credentials only), and other information is intended to be discovered. This type of testing is an interesting compromise in the number of test cases, the cost, the speed, and the scope of testing. Gray-box testing is the most common kind of testing in the security industry.

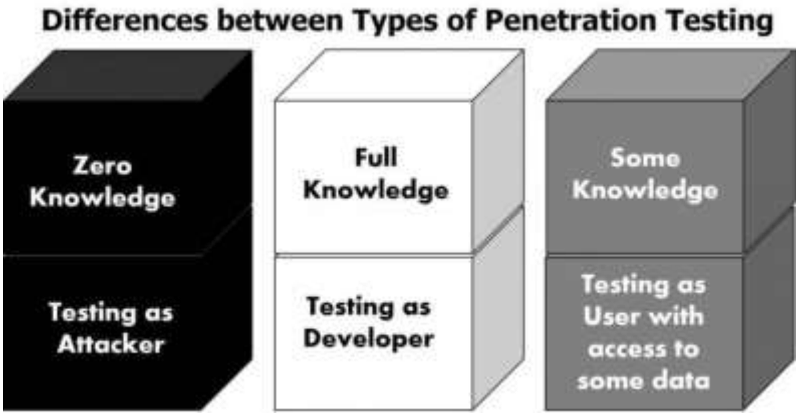

Differences between Types of Penetration Testing
Zero Knowledge
Testing as Attacker
Full Knowledge
Testing as Developer
Some Knowledge
Testing as User with access to some data

Figure 2 - Types of Security Assessments

Even though decompiling on Android is straightforward, the source code may be obfuscated, and de-obfuscating will be time-consuming. Time constraints are therefore another reason for the tester to have access to the source code.

Static Application Security Testing (SAST) involves examining an app's components without executing them, by analysing the source code either manually or automatically. During static analysis, the mobile app's source code is reviewed to ensure appropriate implementation of security controls.

Manual Code Review: The tester performs manual code review by manually analysing the mobile app's source code for security vulnerabilities. Methods range from a basic keyword search via the 'grep' command to a line-by-line examination of the source code. IDEs (Integrated Development Environments) often provide basic code review functions and can be extended with various tools.

DAAC Digital's approach to manual code analysis entails identifying key security vulnerability indicators by searching for certain APIs and keywords, such as database-related method calls like "executeStatement" or "executeQuery". Code containing these strings is a good starting point for manual analysis.

In contrast to automatic code analysis, **manual code review** is very good for identifying vulnerabilities in the business logic, standards violations, and design flaws, especially when the code is technically secure but logically flawed. Such scenarios are unlikely to be detected by any automatic code analysis tool. The security team il investigating the following areas when conducting manual code review:

- Authentication
- Authorization
- Session Management
- Data Storage
- Information Disclosure
- Application Security Issues – XSS, CSRF, SQL Injection, Command Injection, XML Injection, Check Cross Domain Policy, Cookies and others.
- Networking – weak / insecure protocol usage
- Transport Layer Protection – SSL – Encryption-in-transit

Automated Source Code Analysis: Automated analysis tools are used to speed up the review process of Static Application Security Testing (SAST). They check the source code for compliance with a predefined set of rules or industry best practices, then they return a list of findings or warnings and flags for all detected violations. Some static analysis tools will be executed against the compiled app only, some must be fed the original source code, and some run as live-analysis plugins in the Integrated Development Environment (IDE).

Although some static code analysis tools incorporate a lot of information about the rules and semantics required to analyse mobile apps, they can produce false positives. Our security professionals will always review and validate the results.

Dynamic Application Security Testing (DAST) involves examining the app during runtime. This type of analysis is both manual and automatic. It usually doesn't provide the information that static analysis provides, but it is a good way for our team, to detect interesting elements (assets, features, entry points, etc.) from a user's point of view.

The focus of DAST is the testing and evaluation of apps via their real-time execution. The main objective of dynamic analysis is finding security vulnerabilities or weak spots in a program while it is running. Dynamic analysis is conducted both at the mobile platform layer and against the backend services and APIs, where the mobile app's request and response patterns can be analysed.

DAAC Digital uses dynamic analysis to check for security mechanisms that provide sufficient protection against the most prevalent types of attack, such as disclosure of data in transit, authentication and authorization issues, and server configuration errors.

- Application Types – Native, Web Services App (SOAP/REST), Mobile Browser based App, Mobile Hybrid App (Native + Web App)
- Application Mapping – Establishing a baseline for the application, before and after install – file system usage
- Debugging – Examining the application with a debugger attached
- Local Testing – Checking for exposed IPC interfaces – fuzzing, sniffing authentication bypass testing
- Cryptography Testing – Checking for weaknesses with cryptography, brute force key attacks, hard-coded keys / secrets other disclosed information
- Web Application Security Issues – XSS, CSRF, SQL Injection, Command Injection, XML Injection, Check Cross Domain Policy, Cookies etc.
- Authentication – Testing for broken authentication
- Authorization – Weak local filesystem runtime permissions – external configuration manipulation
- File System Analysis – Weak local filesystem runtime permissions – external configuration manipulation
- Memory Analysis
- Remote Application / Server Testing Test discovered backend / hosting / API's – Authentication – Authorization – Session Management – Transport Layer Testing – Server-Side Attacks

Security Assessment Process

Involves the following stages:

Preparation - defining the scope of security testing, including identifying applicable security controls, the organization's testing goals, and sensitive data.

Intelligence Gathering - analysing the environmental and architectural context of the app to gain a general contextual understanding.

This phase involves taking time to learn the target applications purpose and assess it's functionality. This information is then used to correctly scope and assess the level of effort and time required to assess the mobile application.

- Application Type – Application type (mobile web, native, cross-platform)
- Application mapping – manually assessing the application assessing functionality, understanding how the application should function
- Identifying network interfaces, the application uses
- Determining what network protocols are in use
- Determining if the application performs payments processing / commerce transactions and how these are stored
- Determine what hardware is in use – GPS, Bluetooth, TouchID / Camera / Microphone etc.
- Identify any 3rd party library / software / frameworks are in use
- Determine if the application interacts with any other applications
- Assessing server-side information to determine what hosting platforms (AWS, Azure, Rackspace, Heroku etc.) and technologies (Development language, Single Sign On, 2FA, API's) are in use

Mapping the Application - based on information from the previous phases; will be complemented by automated scanning and manually exploring the app. Mapping provides a thorough understanding of the app, its entry points, the data it holds, and the main potential vulnerabilities. These vulnerabilities will be ranked according to the damage their exploitation would cause so that the security tester can prioritize them.

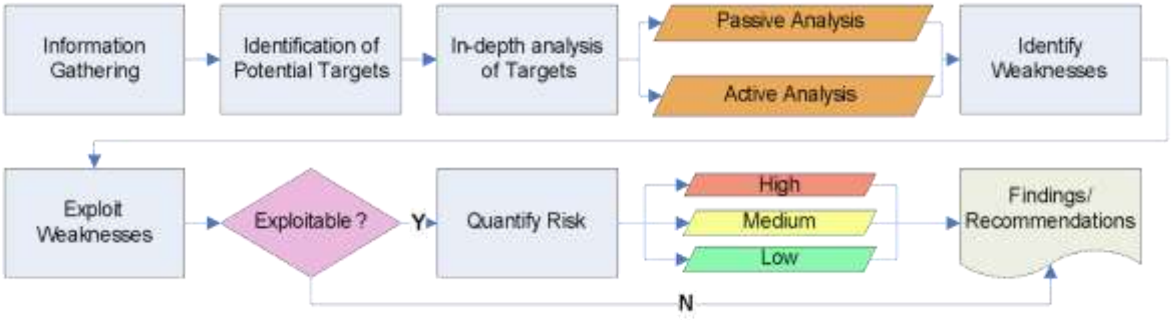


Figure 3 - Mobile Security Assessment Process

Exploitation - in this phase, the security tester tries to penetrate the app by exploiting the vulnerabilities identified during the previous phase. This phase is necessary for determining whether vulnerabilities are real and true positives.

Reporting - in this phase the security tester reports the vulnerabilities. This includes the exploitation process in detail, classifies the type of vulnerability, documents the risk if an attacker would be able to compromise the target and outlines which data the tester has been able to access illegitimately.

Mobile Security Testing Areas

DAAC Digital conducts testing using both automated and manual testing methods. A majority of the testing will focus on manual testing. DAAC Digital uses automated testing, where possible, on back end and web service components, and manually validates identified vulnerabilities. Throughout the assessment DAAC Digital uses a mixture of commercial, open source, and custom-built tools.

In the following sections we'll provide a brief overview of general security testing principles and key terminology that DAAC Digital's security team is investigating in a Mobile Security Assessment:

Authentication

Verifying that Appropriate Authentication is in Place

- Identify the additional authentication factors the app uses.
- Locate all endpoints that provide critical functionality.
- Verify that the additional factors are strictly enforced on all server-side endpoints.

Testing Best Practices for Passwords

Password strength is a key concern when passwords are used for authentication. The password policy defines requirements to which end users should adhere. A password policy typically specifies password length, password complexity, and password topologies. A "strong" password policy makes manual or automated password cracking difficult or impossible.

Testing Stateful Session Management

- Session Management Best Practices
- Session IDs are randomly generated on the server side.
- The IDs can't be guessed easily (use proper length and entropy).
- Session IDs are always exchanged over secure connections (e.g., HTTPS).
- The mobile app doesn't save session IDs in permanent storage.
- The server verifies the session whenever a user tries to access privileged application elements, (a session ID must be valid and must correspond to the proper authorization level).
- The session is terminated on the server side and session information deleted within the mobile app after it times out or the user logs out.

Testing Session Timeout

Minimizing the lifetime of session identifiers and tokens decreases the likelihood of successful account hijacking.

Testing User Logout

The purpose of this test case is verifying logout functionality and determining whether it effectively terminates the session on both client and server and invalidates a stateless token.

Testing Two-Factor, Local Authentication and Step-up Authentication

- One-time password via SMS (SMS-AGE)
- One-time code via phone call
- Hardware or software token
- Push notifications in combination with PKI and local authentication
- Testing Confirm Credentials
- Testing Biometric Authentication
- Testing Keychain Services for Local Authentication

Whatever option is used as 2nd factor, it always must be enforced and verified on the server-side and never on client-side. Otherwise, the 2nd factor can be easily bypassed within the app.

Testing Stateless (Token-Based) Authentication

Token-based authentication is implemented by sending a signed token (verified by the server) with each HTTP request. The most commonly used token format is the JSON Web Token.

We will Identify the JWT library that the server and client use and find out whether the JWT libraries in use have any known vulnerabilities.

- Hashing algorithm
- Token Expiration
- Token Storage
- Information Disclosure
- Tampering with the Hashing Algorithm

Testing OAuth 2.0 Flows

- The user should have a way to visually verify trust (e.g., Transport Layer Security (TLS) confirmation, website mechanisms).
- To prevent man-in-the-middle attacks, the client should validate the server's fully qualified domain name with the public key the server presented when the connection was established.
- On native apps, code grant should be used instead of implicit grant.
- When using code grant, PKCE (Proof Key for Code Exchange) should be implemented to protect the code grant. Make sure that the server also implements it.
- The auth "code" should be short-lived and used immediately after it is received. Verify that auth codes only reside on transient memory and aren't stored or logged.
- Shared secrets should not be used to prove the client's identity because the client could be impersonated. If they do use client secrets, be sure that they are stored in secure local storage.
- Secure the transmission of end-user credentials with a transport-layer method, such as TLS.
- Keep access tokens in transient memory.
- Access tokens must be transmitted over an encrypted connection.

- Reduce the scope and duration of access tokens when end-to-end confidentiality can't be guaranteed or the token provides access to sensitive information or transactions.
- Remember that an attacker who has stolen tokens can access their scope and all resources associated with them if the app uses access tokens as bearer tokens with no other way to identify the client.
- Store refresh tokens in secure local storage; they are long-term credentials.

Testing Login Activity and Device Blocking

Applications should inform the user about all login activities within the app with the possibility of blocking certain devices. Below is a list of common sensitive activities that are usually audited:

- Login attempts
- Password changes
- Personal Identifiable Information changes (name, email address, telephone number, etc.)
- Sensitive activities (purchase, accessing important resources, etc.)
- Consent to Terms and Conditions clauses

Network Communication

Intercepting HTTP(S) and Non-HTTP Traffic.

By monitoring the requests between the mobile app client and the backend, the security team can easily map the available server-side APIs and gain insight into the communication protocol.

- Simulating a Man-in-the-Middle Attack with Access Point
- Setting a Proxy Through Runtime Instrumentation on a rooted or jailbroken device
- Verifying Data Encryption on the Network (strong Cipher suites)
- Verifying that Critical Operations Use Secure Communication Channels
- Testing Endpoint Identify Verification
- Testing Custom Certificate Stores and Certificate Pinning
- Testing the Network Security Configuration Settings
- Testing the Security Provider
- Testing App Transport Security (ATS)
- Testing Custom Certificate Stores and Certificate Pinning

Cryptography in Mobile Apps

Cryptography plays an especially important role in securing the user's data - even more so in a mobile environment, where attackers having physical access to the user's device is a likely scenario.

DAAC Digital verifies the encryption algorithms that converts plaintext data into cipher text that conceals the original content.

- Symmetric-key encryption algorithms
- Public-key encryption algorithms
- Hashing functions
- Message Authentication Codes (MACs)

- Signatures
- Key Derivation Functions (KDFs)

Testing Encryption and Common Cryptographic Configuration Issues

The security of symmetric encryption and keyed hashes (MACs) depends on the secrecy of the key. If the key is disclosed, the security gained by encryption is lost.

The security team verifies that no keys, certificates or passwords are stored within the source code.

- The password to the client certificate isn't stored locally or is locked in the device Keychain.
- The client certificate isn't shared among all installations.
- Encrypting locally stored data with a static, hardcoded encryption key and compiling that key into the app.
- The security team verifies if there are custom Implementations of Cryptography. Inventing proprietary cryptographic functions is time consuming, difficult, and likely to fail.
- Insufficient Key Length
- Weak Key Generation Functions
- Inadequate AES Configuration
- Predictable Initialization Vector
- Padding Oracle Attacks due to Weaker Padding or Block Operation Implementations
- Identifying Insecure and/or Deprecated Cryptographic Algorithms
- Cryptography Reuse
- Weak Random Number Generators

Code Quality

Common vulnerabilities such as SQL injection, buffer overflows, and cross-site scripting (XSS), may manifest in apps when neglecting secure programming practices. The same programming flaws may affect both Android and iOS apps to some degree.

Bellow there is an overview of the most common vulnerability generated by bad programming that the security team will try to identify:

- Injection Flaws: SQL Injection, XML Injection, Cross-Site Scripting Flaws (XSS)
- Identifying possible entry points for untrusted input then tracing from those locations to see if the destination contains potentially vulnerable functions.
- Identifying known, dangerous library / API calls (e.g. SQL queries) and then checking whether unchecked input successfully interfaces with respective queries.
- Memory Corruption Bugs: Buffer Overflows, Out-of-bounds-access, Dangling pointers, Use-after-free, Integer overflows, Format string vulnerabilities

- Making Sure That the App is Properly Signed
- Testing Whether the App is Debuggable
- Testing for Debugging Symbols
- Testing for Debugging Code and Verbose Error Logging
- Checking for Weaknesses in Third Party Libraries
- Testing Exception Handling
- Test That Free Security Features Are Activated

Tampering and Reverse Engineering

Reverse engineering a mobile app is the process of analyzing the compiled app to extract information about its source code. The goal of reverse engineering is comprehending the code. The following techniques are used:

- Disassembling and Decompiling
- Review decompiled code
- Sandbox Inspection on rooted device
- Debugging Release Apps
- Execution Tracing
- Patching, Repackaging, and Re-Signing

Tampering is the process of changing a mobile app (either the compiled app or the running process) or its environment to affect its behavior. For example, an app might refuse to run on our rooted test device, making it impossible to run some of the tests. In such cases, our team will alter the app's behavior.

Basic Tampering Techniques used:

- Binary Patching
- Code Injection
- Dynamic Binary Instrumentation
- Emulation-based Dynamic Analysis
- Program Analysis with Symbolic/Concolic Execution

Android Security Assessment

The Android application attack surface consists of all components of the application, including the supportive material necessary to release the app and to support its functioning. When assessing an Android Application, the security team takes into consideration the following:

- Android architecture

- Android security
- System-wide security
- Software isolation
- Network security
- Anti-exploitation
- Android application structure: Activities, Fragments, Intents, Broadcast receivers, Content providers and services
- Android application publishing: Signing Process, APK Signing Schemes, Publishing Process
- Android application attack surface

Data Storage on Android

Protecting authentication tokens, private information, and other sensitive data is key to mobile security. The following list of persistent storage techniques are widely used on the Android platform:

- Shared Preferences
- SQLite Databases
- Firebase Databases
- Realm Databases
- Internal Storage
- External Storage
- Keystore

In addition to this, there are a number of other functions in Android built for various use cases that can also result in the storage of data and respectively should also be tested, such as:

- Logging Functions
- Android Backups
- Processes Memory
- Keyboard Caches
- Screenshots

The security team will focus on finding vulnerabilities and executing the following tests:

- Testing Local Storage for Sensitive Data
- Testing Local Storage for Input Validation
- Testing Logs for Sensitive Data
- Determining Whether Sensitive Data Is Shared with Third Parties

- Determining Whether the Keyboard Cache Is Disabled for Text Input Fields
- Determining Whether Sensitive Stored Data Has Been Exposed via IPC Mechanisms
- Checking for Sensitive Data Disclosure Through the User Interface
- Testing Backups for Sensitive Data
- Finding Sensitive Information in Auto-Generated Screenshots
- Checking Memory for Sensitive Data

Testing the Device-Access-Security Policy

Android Cryptographic APIs

The security team will verify the strength and usage of security keys in Android:

- Testing Symmetric Cryptography
- Testing the Configuration of Cryptographic Standard Algorithms
- Testing the Purposes of Keys
- Testing Random Number Generation

Android Platform APIs

Android assigns a distinct system identity (Linux user ID and group ID) to every installed app. Because each Android app operates in a process sandbox, apps must explicitly request access to resources and data that are outside their sandbox. They request this access by declaring the permissions they need to use system data and features. Depending on how sensitive or critical the data or feature is, the Android system will grant the permission automatically or ask the user to approve the request.

- Testing App Permissions
- Activity Permission Enforcement
- Testing for Injection Flaws
- Testing for Fragment Injection
- Testing for URL Loading in WebViews
- Testing Custom URL Schemes
- Testing for Sensitive Functionality Exposure Through IPC
- Testing JavaScript Execution in WebViews
- Testing WebView Protocol Handlers
- Determining Whether Java Objects Are Exposed Through WebViews
- Testing Object Persistence
- Testing for Overlay Attacks

- Testing enforced updating

Android Anti-Reversing

In the context of anti-reversing, the goal of root detection is to make running the app on a rooted device a more difficult, which in turn blocks some of the tools and techniques reverse engineers like to use. The security team will test the efficiency of the anti-reversing measures and will attempt to bypass them in various methods:

- Renaming binaries
- Unmounting /proc to prevent reading of process lists
- Using Frida or Xposed to hook APIs on the Java and native layers
- Hooking low-level APIs by using kernel modules.
- Patching the app to remove the checks.
- Bypassing Debugger Detection
- Bypassing File Integrity Checks
- Bypassing Reverse Engineering Tools Detection
- Bypassing Emulator Detection
- Bypass Runtime Integrity Checks
- Testing Obfuscation levels
- Testing Device Binding

iOS Security Assessment

The iOS application attack surface consists of all components of the application, including the supportive material necessary to release the app and to support its functioning. When assessing an iOS Application, the security team takes into consideration the following:

- iOS security architecture
- iOS application structure
- Inter-process Communication (IPC)
- iOS application publishing
- iOS Application Attack Surface
- Hardware Security
- Secure Boot
- Code Signing
- Sandbox
- Encryption and Data Protection

- General Exploit Mitigations
- App Permissions

Data Storage on iOS

The protection of sensitive data, such as authentication tokens and private information, is key for mobile security.

- Testing Local Data Storage: Data Protection API
- Checking Logs for Sensitive Data
- Finding Sensitive Data in the Keyboard Cache
- Determining Whether Sensitive Data Is Exposed via IPC Mechanisms
- Checking for Sensitive Data Disclosed Through the User Interface
- Testing Backups for Sensitive Data
- Testing Auto-Generated Screenshots for Sensitive Information
- Testing Memory for Sensitive Data

iOS Cryptographic APIs

The security team attempts to identify the use of Cryptographic APIs in the source code and interpret cryptographic configurations.

- Verifying the Configuration of Cryptographic Standard Algorithms
- Testing Key Management
- Testing Random Number Generation

iOS Platform permissions

In contrast to Android, where each app runs on its own user ID, iOS makes all third-party apps run under the non-privileged mobile user. Each app has a unique home directory and is sandboxed, so that they cannot access protected system resources or files stored by the system or by other apps.

Even though Apple urges to protect the privacy of the user and to be very clear on how to ask permissions, it can still be the case that an app requests too many of them for non-obvious reasons.

- Testing for Sensitive Functionality Exposure Through IPC
- Testing Custom URL Schemes
- Testing iOS WebViews
- Testing WebView Protocol Handlers
- Determining Whether Native Methods Are Exposed Through WebViews
- Testing Object Persistence
- Testing enforced updating

Bypassing iOS Anti-Reversing Defenses

Jailbreak detection mechanisms are added to reverse engineering defense to make running the app on a jailbroken device more difficult. This blocks some of the tools and techniques reverse engineers like to use. The security team will test the efficiency of the anti-reversing measures and will attempt to bypass them in various methods:

- Testing Anti-Debugging Detection
- Bypass File Integrity Checks
- Testing Reverse Engineering Tools Detection
- Testing Obfuscation
- Device Binding

Tools used as part of the Mobile Assessment Methodology:

- MobSF – DAST and SAST iOS and Mobile
- BurpSuitePro – Traffic manipulation iOS and Android
- JDB – for debugging
- ADB – for interaction with Android
- Drozer – Android Penetration testing framework
- Xposed – on client-side Android
- Frida – for iOS and Android run-time manipulation
- Cydia Substrate modules – for iOS
- iFunBox – android debugging/reverse engineering
- class-dump – examining Objective-C Run-time iOS
- cycript – iOS run-time manipulation
- Introspy-iOS – run-time assist manipulation
- Keychaindumper – iOS dumping stored encrypted keys
- OWASP Zap – alternative to Burp Suite
- X-con – bypassing jailbreak detection
- tsProtector – bypassing jailbreak detection
- Clutch – executable dumper
- Sqlite3 – navigate/read/write into .db files
- Burp Suite Mobile Assistant – facilitate testing of iOS
- Wireshark – packet analyzer

- Fridump – Frida framework to dump accessible memory addresses
- Qark – analyze several security related Android vulnerabilities (source code or packaged)
- Jadx – Dex to java decompiler Android
- APKTool – Reverse Engineer Android
- Needle – security assessment on iOS
- Objection - run-time mobile exploration toolkit iOS and Android

List of Tools

Below is the list of primary tools that are used during the Penetration Testing phases.

- **Nessus and Nexpose** Vulnerability Scanners
- **OpenVas** Vulnerability Scanner
- **Burp** Professional
- **Acunetix** Professional
- **NetSparker** Professional
- **Gobuster**, **Dirbuster**, **Filibuster** directory brute-force tools
- **Nikto** web vulnerability Scanner
- **THC Hydra** brute force tool
- **Nmap**, Nmap NSE
- **Kali** distribution

Below is the list of supporting tools and scripts that may be used depending on a specific scenario found and employed across testing cycles (non-exhaustive).

- **Achilles –** A tool designed for testing the security of web applications
- **ActivePerl –** PERL for Windows
- **Acunetix –** Web app vulnerability scanner
- **ADMft –** An FTP brute-force tool
- **ADMpop –** A POP brute-force tool
- **ADMsmb –** An SMB brute-force tool
- **ADMsnmp –** An SNMP brute-force tool
- **Airsnort –** A wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered
- **Arirang –** A powerful webserver security scanner (according to their own description)
- **Borg –** A Free disassembler for Win32 PE files
- **Brutus –** An Windows GUI brute-force tool for FTP, telnet, POP3, SMB, HTTP, etc

- **Burp Proxy** - Burp Proxy is an intercepting proxy server for security testing of web applications.
- **Cain and Abel** – GUI password sniffer for Windows
- **Checkmarx** – Source code review
- **Chntpw** – An image of a (Linux) boot disk that allows changing any password on a Windows NT/2000.
- **Cisco TFTPServer** – A free TFTP Server.
- **CiscoCFG** – Example PERL script that could be used to upload/download config files to/from Cisco routers using SNMP
- **CiscoCrack** – PERL script that unscrambles Cisco level 7 passwords
- **ciscosnmpdos** – Example script to test SNMP DOS vulnerability in Cisco IOS. Payload taken from OULU University SNMP test suite. Using this on a vulnerable router will make it reboot. Use with care!
- **Cishttpex.pl** – Perl script that finds the magic number for the HTTP vulnerability
- **CmdAsp.asp** – An ASP page that allows executing commands on a server
- **Crack** – A password cracker
- **CrypTool** – A Cryptanalysis tool.
- **cURL** – Curl is a tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE and LDAP
- **DCEtest** – This little utility dumps MSRPC endpoint information from Windows systems.
- **DumpEvt** – Dump the event log in a format suitable for importing into a database
- **DumpReg** – Dumps the registry, making it easy to find keys and values containing a string
- **DumpSec** – Dumps the permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers and shares
- **ELSave** – ELSave is a tool to save and/or clear a NT event log
- **Elza** – A family of tools for arbitrary HTTP communication with picky web sites for the purpose of penetration testing and information gathering
- **Enum** – A tool to enumerate, using null and user sessions, Win32 (NT) information
- **EPDump** – A little tool to dump the contents of the endpoint mapper
- **Ethereal** – Ethereal is a free network protocol analyser for UNIX and Windows.
- **Ettercap** – Ettercap is a multipurpose sniffer/interceptor/logger for switched LAN. It supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis.
- **FScan** – A command-line port scanner. Supports TCP and UDP
- **GetAccount** – Windows remote user enumerator based on the SID2USER calls
- **GetPass!** – Tools that unscrambles Cisco level 7 passwords
- **Grinder** – Tool that scans a range of IP address to query web servers for a given document
- **Hackman** – Hackman is a freeware hex editor and disassembler
- **HPing** – HPing is a command-line oriented TCP/IP packet assembler/analyser. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features.
- **Hyena** – Centralised Windows domain management tool. Very useful when auditing 'internal' networks.
- **ICMPush** – ICMPush is a tool that sends ICMP packets fully customised from command line
- **IDA Pro** – Demo (IDA Pro 4.21) and freeware version (ida37fw) of the popular (commercial) disassembler.

- **IEHistory** – The Internet Explorer History Viewer will parse and print the history of URL's visited using the Microsoft Internet Explorer version 3.x, 4.x and 5.x.
- **IIS5-Koei** – Graphical exploit for the IIS5 .printer ISAPI buffer overflow.
- **IISCat** – IIS FPSE bug that allows to view ASP script source code
- **IISCrack** – This ISAPI DLL allows you to gain SYSTEM level access to an IIS 5.0 system.
- **IISHack** – An exploit for a buffer overflow vulnerability in IIS4
- **Irpas** – Internetwork Routing Protocol Attack Suite
- **Jill** – Exploit for the .PRINTER buffer overflow in IIS5
- **John The Ripper** – A password cracker
- **Joshua** – Perl based war dialer
- **L0phtcrack** – NTLM/Lanman password auditing and recovery application (read - cracker)
- **LANguard network scanner** – Port & vulnerability scanner
- **Legion** – This is a Win32 file share scanner
- **Lsadump** – An application to dump the contents of the LSA secrets on a machine, provided you are an Administrator
- **MingSweeper** – A network reconnaissance tool capable of performing Ping sweeps, Reverse DNS sweeps, TCP and UDP port scans, OS identification and application identification
- **msadc** – The famous exploit for th2e RDS vulnerability in IIS4
- **Msn666** – MSN666 is a simple sniffing program against msn messenger, it intercepts all msn messages on your network, so you could find who is on the net with msn messenger on and who talk with whom.
- **Metasploit** – Exploitation framework
- **Nbtscan** – A program for scanning IP networks for NetBIOS name information
- **NBTStat** – This is a small Unix utility that does the equivalent of NT's nbtstat
- **NetBIOS auditing tool** – Performs various security checks on remote servers running NetBIOS file sharing services
- **Netcat** – The swiss army knife of network tools. A simple utility which reads and writes data across network connections, using TCP or UDP protocol
- **netddemsg** – Privilege escalation exploit for Windows 2000 (up to and including Service Pack 1)
- **NetE** – Null session enumeration tool.
- **NetSed** – NetSED is small and handful utility designed to alter the contents of packets forwarded thru your network in real time.
- **Netstumbler** – Wardriving software.
- **Netu** – Bruteforce password tester using WNetAddConnection2() or RAS.
- **Netviewx** – A tool that lists servers in a domain or a workgroup
- **Nexpose** – A vulnerability management solution which analyses vulnerabilities, controls, and configurations to find the who, what, and where of IT security risk.
- **Nikto** – A web server scanner, based on and inspired by Whisker 1.4. Support for proxy, host authentication, and SSL
- **Nltest** – NLTEST.EXE is a very powerful command-line utility that can be used to test Trust relationships and the state of Domain Controller replication in a Microsoft Windows NT Domain.
- **NMAP** – The best known port scanner around.
- **N-Stealth** – A nice graphical CGI scanner. The software comes with a extensive database of over 19,000 vulnerabilities and exploits.

- **OpenSSL** – The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library.
- **p0f** – Passive OS Fingerprinting - A tool that listens on the network and tries to identify the OS versions from the information in the packets.
- **Pwdump** – Tools that grab the hashes out of the SAM database, to use with a brute-forcer like L0phtcrack or John
- **RPCTools** – The RPC tools package contains three separate tools for obtaining information from a system that is running RPC services
- **Samba** – Samba is an Open Source/Free Software suite that provides seamless file and print services to SMB/CIFS clients
- **Samba-TNG** – Samba-TNG is an Open Source/Free Software suite that implements a dce/rpc* library
- **Samdump** – Dumping passwords without being an administrator
- **SamSpade** – Graphical tool that allows to perform different network queries - ping, nslookup, whois, IP block whois, dig, traceroute, finger, SMTP VRFY, web browser keep-alive, DNS sone transfer, SMTP relay check,etc.
- **Sara** – A security analysis tool based on the SATAN model. It is updated twice a month to address the latest threats
- **Satan** – Security Administrator Tool for Analysing Networks
- **ScanDNS** – Script that scans a range of IP addresses to find DNS names
- **ScanSSH** – An SSH version scanner
- **ShoWin** – Show information about windows, reveal passwords, etc.
- **SID2User** – SID2User and User2SID are command line interfaces to WIN32 functions, LookupAccountName and LookupAccountSid
- **Sing** – Send ICMP Nasty Garbage. A little tool that sends ICMP packets fully customised from command line
- **Skravel** – Windows remote user enumerator based on the SID2USER calls
- **SMBGrind** – Tool to brute-force SMB shares over the network (part of CyberCOP)
- **SolarWinds** – Network Management & Discovery Tools
- **Sqlat** – SQL Auditing Tools
- **SqlBf** – MSSQL server brute force tool
- **SqlDict** – MSSQL server dictionary attacker
- **SqlExec** – A little tool that allows to execute commands on an MSSQL server using the XP_CMDSHELL stored procedure
- **SqlPing** – SQLPing is a utility for querying SQL Servers (2000+) listening on UDP 1434 to return detailed information about the instances installed. Note that broadcast addresses may return multiple results.
- **SqlPoke** – Used to scan a range of IP addresses for SQL Servers and then execute a predefined script. Could be used to track down SQL Servers in your own organisation and ensure they stay locked down
- **SqlTools** – Two Unix tools that allow to hack MSSQL servers
- **SSHWinClient** – A full blown Win32 SSH Client implementation
- **SSLProxy** – A tool that allows running non SSL-aware tools/programs over SSL.
- **Strobe** – A command-line port scanner that also performs banner grabbing

- **STunnel –** Stunnel is a program that allows you to encrypt arbitrary TCP connections inside SSL
- **Superscan –** Simple TCP port scanner with nice GUI. Also performs banner grabbing.
- **Teleportpro –** Software that mirrors websites to your hard disk (Evaluation Version)
- **THC-scan –** A nice war dialer
- **twwwscan –** A fast windows based command line WWW Vulnerability scanner
- **Typhon –** Evaluation version of the commercial vulnerability scanner
- **UCD-Snmp (aka NET-Snmp) –** Various tools relating to the Simple Network Management Protocol including snmpget, snmpwalk and snmpset.
- **Unix Exploits –** Various publicly available exploits
- **UserDump –** Tool that uses RPC mechanisms (LookupAccountSid, LookupAccountName, and NetUserGetInfo) to enumerate users on NT. Bypasses the RestrictAnonymous registry setting
- **Vlad –** An open-source security scanner that checks for the SANS Top Ten security vulnerabilities
- **WAST –** Microsoft Web Application Stress - A tool for stress testing web servers
- **Webreaper –** Software that mirrors websites to your hard disk (Freeware)
- **Wellenreiter –** Wellenreiter is a GTK/Perl program that makes the discovery and auditing of 802.11b wireless networks much easier. All three major wireless cards (Prism2 , Lucent, and Cisco) are supported.
- **Whisker –** The most famous CGI scanner
- **WinDbg –** The WinDbg debugger is a powerful, graphical tool that allows you to debug applications on Microsoft® Windows NT® and Microsoft Windows®.
- **Windows exploits –** Various Windows exploits
- **WinPcap –** Packet capture device drivers for Windows. Most sniffing utilities use these.
- **WinScan –** Tool that scans all systems in a domain and enumerates shares, users, etc.
- **WPoison –** Find any potential SQL-Injection vulnerabilities in dynamic web documents which deals with databases: php, asp, etc.

Threat Classification and Reporting

During our security assessments or penetration testing, when any exploitable vulnerability is discovered, further research is conducted on that vulnerability to identify its level of severity. For example, a remote exploit which can provide a root or super-user session on the targeted server is classified as a high risk finding; whereas information disclosure regarding an internal hostname would be classified as a low risk finding. The risk is calculated according to the following criteria:

Impact

The security impact on your web application in the event of an exploitation of this vulnerability by an attacker. This criterion indicates the benefit of the attack to the attacker.

Ease of Exploitation

The level of difficulty for an attacker to exploit this problem. Difficulty could increase due to technical complexity, the need for prior knowledge of the network, or other factors. This criterion indicates the cost in time and resources of the attack for the attacker.

Popularity and Ease of Identification of the Vulnerability

This criterion factors in the public availability of information and tools to detect the vulnerability. Problems that are exploited by Internet worms or that have easy to use exploit code available on the Internet, for example, would get a higher rating. This criterion indicates the probability of an attack.

The risk is classified as follows:

Risk Classification	Characteristics
Critical Risk	Vulnerabilities in this category usually have the following characteristics: Exploitation of the vulnerability results in root/administrator-level access to the system; The information required in order to exploit the vulnerability, such as example code, is widely available to attackers; Exploitation is usually straightforward, in the sense that the attacker does not need any special authentication credentials or knowledge about individual victim systems, and does not need to persuade a target user, for example via social engineering, into performing any special functions.
High Risk	Vulnerabilities that score in the high range usually have the following characteristics: The vulnerability is difficult to exploit; Exploitation does not result in elevated privileges, but may grant unintended access to data; Exploitation does not result in a significant data loss.

Medium Risk	Vulnerabilities that score in the medium range usually have the following characteristics: Denial of service vulnerabilities that are difficult to set up; Exploits that require an attacker to reside on the same local network as the victim; Vulnerabilities that affect only nonstandard configurations or obscure applications; Vulnerabilities that require the attacker to manipulate individual victims via social engineering tactics; Vulnerabilities where exploitation provides only very limited access.
Low Risk	Vulnerabilities in the low range typically have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access.
Informational	These are not vulnerabilities, but additional information gleaned from the target during vulnerability testing.

Project Deliverables

At the conclusion of each phase, DAAC Digital will provide written documentation of the approach, findings and recommendations associated with the project. The documentation will consist of the following:

Detailed Technical Reports

The reports will be clear enough for Agenția de Guvernare Electronică (AGE) Security Team or any other skilled security tester to re-perform the test scenario. All bulk data (like Vulnerability scan results) will be put in MS Excel format. The main document will include:

- The methodology employed
- Tools used (if applicable)
- Positive security aspects identified (if applicable)
- Detailed technical vulnerability findings
- Supporting evidences, screenshots or POCs (if applicable)
- Executive and Management Summary
- Methodology section
- Individual Risk rating based on Inherent and Residual Risk
- An assignment of a risk rating for each vulnerability
- Proposed remediation steps
- Any other recommendations and conclusions of overall security posture (if applicable)
-
-

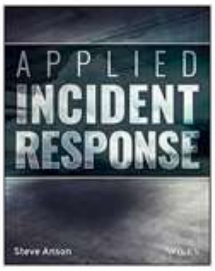
Delivery Team

DAAC Digital can involve the experienced team to deliver the project. The final consultant's assignment and task allocation will be decided during projects initiation based on actual project needs.

Along with our practical experience, our team maintains a range of industry certifications including:

GWAPT
GXPN
OFFENSIVE security OSCP
NEXPOSE ADVANCED CERTIFIED ADMINISTRATOR
GCIH
GPEN
CISSP
MICRO FOCUS Certified Expert ArcSight Analyst
EnCASE CERTIFIED EXAMINER EnCE OPENTEXT
GDAT
CISM
GCFA
LEAD AUDITOR ISO 27001 Information Security Management System
cellebrite certified CCME mobile examiner

We have also authored industry-leading IT security books:



Scope of Work

Scope Description

DAAC Digital will perform different types of assessments, in different cycles. The type of assessments will be discussed and agreed upon (as part of the timelines) with the client, before initiating the engagement.

DAAC Digital proposes to conduct Security Services divided into phases as described below:

- **Vulnerability Testing:** Automated and manual scanning to identify known vulnerabilities, misconfigurations, and missing security updates.
- **Penetration Testing:** Simulated real-world attacks using:
    - Black Box
    - Gray Box
    - White Box
- **Security Code Review:** Manual and static analysis of source code to identify programming defects, potential vulnerabilities, and business logic errors.

- **Security Audit Services (Defensive / GAP Assessment)**

    - Verification of system/service compliance with applicable security standards and best practices, including eIDAS requirements for trust services and digital identity wallets (EUDI Wallet).

- **Methodology and Standards Applied**

    - OWASP standards (WSTG, MASVS, MASTG, OWASP Top 10)
    - ISO/IEC standards (27001/27002/27005, 30107 for biometric systems)
    - NIST SP 800-115 for planning, execution, and documentation of security tests
    - Other methodologies as needed: MITRE ATT&CK, CIS Benchmarks, PTES

Governance and Communication Model

While conducting this assessment, the Agenția de Guvernare Electronică (AGE) team will be part of the project team and is the technical lead when assessing the infrastructure. The details of the governance model and method of communication is to be discussed and defined in the kick-off meeting.

Project Timeline

The final schedule will be agreed upon during the project initiation. During the execution phase, the project plan might be adjusted to adapt the situation and respond to identified issues. At this stage, it is not possible to build a detailed project plan since the actual duration will depend on test readiness on the Agenția de Guvernare Electronică (AGE) side.

Resource Mobilization

DAAC Digital offers services to be mobilized within a minimum period of **10-15 business days** from official commencement. However, maintaining continuous communication between the teams and early information sharing about plans for individual components can significantly reduce the lead time for the resource's availability.

Assumptions and Exclusions

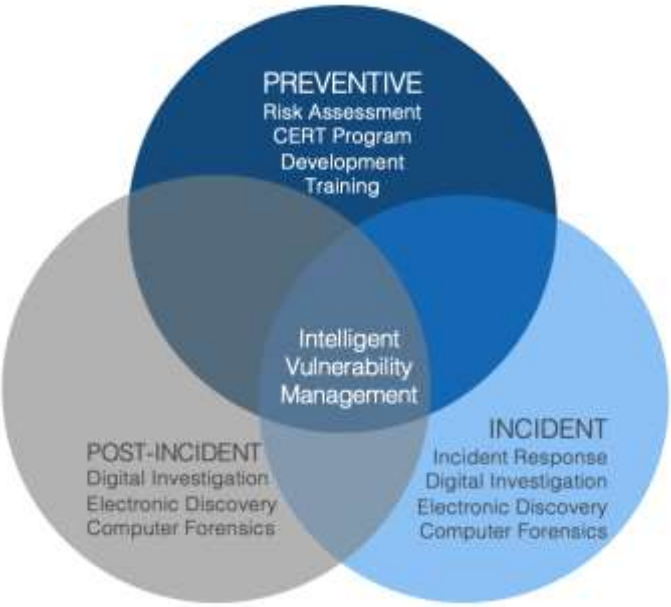
- All prerequisites will be defined by DAAC Digital before starting the assessment. Agenția de Guvernare Electronică (AGE) will be responsible for fulfilling all prerequisites according to the defined schedule.
- Any documents, network diagrams, configuration files, user accounts, etc. have to be provided before the start of the tasks where these documents/information are needed. DAAC Digital will provide a secure way of exchanging sensitive information.
- DAAC Digital will provide remediation and/or recommendations but will not be responsible for fixing/mitigating gaps and vulnerabilities.
- Appropriate backups of applications, databases, etc. should be taken by Agenția de Guvernare Electronică (AGE) before the start of the penetration tests.
- Each activity included within the proposal will be time-based and whenever is needed, the security team from DAAC Digital side together with the Agenția de Guvernare Electronică (AGE)`s team will prioritize the assets and testing scenarios to get the best possible output.

  The list of IPs in the scope of the tests must be approved by Agenția de Guvernare Electronică (AGE) In the case of network segments, list of excluded IPs must be provided by Agenția de Guvernare Electronică (AGE).

Company Overview

DAAC Digital provides information security services to counter the new generation of threats. Our team combines years of security testing and consulting experience throughout the Middle East and around the world with expertise acquired working for different industries and government agencies. We understand the emerging and evolving threat landscape and have years of experience advising an elite group of clientele for which their IT security is a top priority.

CT Defense exposes threats to information, both proactively and reactively, with the single purpose of strengthening our clients' defenses. And while our preference is to expose vulnerabilities before attackers do, our expertise is all too often applied after the fact.



Vulnerabilities, either disclosed or undisclosed, exist in all information technology systems. A dedicated attacker, with enough time and resources, can and will penetrate even the most secure network. The key is to manage this risk. DAAC Digital does so through intelligent vulnerability management.

Intelligent vulnerability management begins with preventive measures. DAAC Digital 's preventive services are built around the fact that an organisation's defences must be resilient, multi-layered, and reactive to attack. Our penetration testing, security architecture review, and vulnerability assessment services provide the starting point for designing and implementing a solution that will address both today's attacks and the emerging threats of tomorrow. Our training and human resourcing services serve to provide our clients with the knowledge transfer and skills updates needed to keep pace with the rapidly evolving threat landscape, and our policy and procedure development services ensure that proper planning and organizational systems are in place to address contingencies that will eventually arise.

When a network has been breached, either due to inadequate planning or through the use of undisclosed vulnerabilities by a dedicated attacker, incident response protocols are implemented.

Our incident response services borrow from the concept of a forward medical team in battlefield conditions. In these circumstances, DAAC Digital takes point on rapidly deploying and beginning the important work of discovering and addressing the sources of harm and the extent of damage to complex information systems in a nimble, aggressive manner. Incident response, digital investigations, and computer forensics all fall into this stage of intelligent vulnerability management.

Managing vulnerabilities requires the ability to constantly evaluate and understand potential attacks and adapt to them. After any incident, the data collected during the response provides immense value by allowing you to see what failed, why it failed, and what techniques the attacker employed against your defenses. Analysis of this data allows the feedback loop to be closed, as the information gained by an effective response is fed back into the preventive measures and used to enhance the network defenses.

DAAC Digital takes an approach to security that starts with one harsh reality: if your data is valuable enough, it can be stolen. The challenge is to make it as difficult as possible for an attacker, to design systems that not only detect and deter known attacks but also that detect breaches in security by unknown exploits, and to have a system in place to quickly react to any breach in a manner that minimizes the damage.

Our clients benefit from this approach as our mindset is geared toward holistic management of threats. We know that investments in securing your information must be allocated wisely, and our vulnerability management provides guidance to our clients on these crucial decisions. Our blended experience in information security, digital forensics, incident response and human resource training uniquely positions us to be your trusted security advisor.

References

The projects listed are examples of successfully completed deliveries:

Organisation	Project Type
Du (UAE)	Network Security Review and VAS Security Risk Assessment
ADIB (Qatar and UAE)	Penetration Testing Services for Infrastructure and Applications Red Teaming engagements
ENBD (UAE)	Comprehensive security audit for containerized environment Infrastructure Control Reviews Annual Security Assessments
Asiacell (Iraq)	Web and Application Security Assessment
Telecommunications Regulatory Authority (TRA) (UAE)	Development of a Security Assessment Web Portal, which regularly evaluates the security posture of all government web sites and tracks trends
Network International (NI)	Core network Security Assessment and Penetration Testing on different applications
DIFC	Core network Security Assessment and Penetration Testing
PalTel	IP MPLS Strategic Security Audit including review of DDOS risks
Mobily (Saudi Arabia)	Mobile Core Security Review - including detailed technical testing and Penetration Testing
Zain (Saudi Arabia)	Mobile Core Security Audit and VAS Security Review
STC (Saudi Arabia)	Telecoms Security Requirements and Detailed Security Design and MBSS Development Triple AAA Security Requirements
CellC (South Africa)	Technology Risk Review and Business Impact Analysis Information Security Policy Support Business Support Systems Technical Risk Review

Contact Us

For questions relating to this document, please contact:

Pusoiu Andrei, *Director*

E: info@dsi.md
M: +40752-607.204

Our Office

DAAC Digital Equipment and Services

info@dsi.md

Calea Ieşilor 10, Chişinău 2069, Moldova

IQCS ASCB(E) IRQAO
ISO 9001:2015

THANK YOU !