QUALYSEC
BEYOND CYBERSECURITY

METHODOLOGY

IOT PRODUCT SECURITY REVIEW

IOT PRODUCT
SECURITY REVIEW

Qualysec's product security review methodology uses advanced hardware and software security assessment techniques to thoroughly evaluate products and their associated infrastructure and systems. The process starts by identifying potential threats to the system, taking into account factors such as the operating environment, users, and the sensitivity of the data processed. Based on this information, the assessment team develops an attack plan targeting areas that are likely to be of interest to attackers. The team then uses both past experience and the latest security research to test the system using a variety of attack techniques. Any issues discovered are then evaluated to determine their impact on the product, organization, and its customers. A product security review, when combined with an application penetration test of client-owned cloud applications and services, can help ensure the security and privacy of products and data.

PROCESS OVERVIEW

Phase 1	Phase 2	Phase 3	Phase 4
PRE-ASSESSMENT	INFORMATION GATHERING & AUTOMATED TESTING	MANUAL PRODUCT TESTING & CODE ANALYSIS	ANALYSIS & REPORTING
	APPLICATION SCANNING FUZZ TESTING THREAT IDENTIFICATION	SCANNING VALIDATION CODE ANALYSIS SOFTWARE TESTING & ANALYSIS HARDWARE TESTING & ANALYSIS	

PHASE 1 : PRE-ASSESSMENT

The following prerequisites must be met to ensure the efficient and successful completion of the assessment.

PRE-ASSESSMENT REQUIREMENTS	
PRODUCT DOCUMENTATION	The assessment team needs comprehensive documentation, such as: • Product documentation, including manuals, user guides, setup instructions, FAQs, product information, platform references, technical data, and whitepapers • Business, technical, and functional specifications • Diagrams related to infrastructure and architecture • API references with sample code • Network or application protocol specifications • Database schemas, LDAP models, and other back-end data storage documentation • Server configuration details • Any other relevant documentation related to the product • Completed scoping survey.
ENVIRONMENT ACCESS	The assessment team may need access to the following resources pertaining to the product's deployment environment: • Information about the servers, operating systems, middleware, firmware, network or application access, and any third-party dependencies • Keys or passphrases are required to access the hardware or software.
CREDENTIALS	The assessment team may require access to the following application resources, such as: • At least two sets of credentials for each role • Any information necessary to use the product (e.g., password reset information, security-question answers, and secure tokens)

PRE-ASSESSMENT REQUIREMENTS

SOURCE CODE	The assessment team may require access to the product's source code, including: • The complete, build-quality source code • Pre-compiled, functional binaries • Any third-party or related libraries used in the product software • Access to the product software build environment • Firmware images • Any utilities, tools, or test harnesses.
HARDWARE	The assessment team may need access to the following resources pertaining to the product's deployment environment: • Information about the servers, operating systems, middleware, firmware, network or application access, and any third-party dependencies • Keys or passphrases are required to access the hardware or software.
DUE CARE	Qualysec strives to minimize network downtime during assessments by carefully planning and executing automated scanning, manual validation, and penetration testing. Before testing begins, the assessment team consults with the client to identify potential risks and establishes a clear plan for addressing any disruptions that may occur.
AUTHORITY	Before testing begins, consent must be obtained from any third-party hosts for any part of the product or associated resources.

PHASE 2 :
INFORMATION GATHERING & AUTOMATED TESTING

The assessment team starts by analyzing the system's performance during regular usage through automated testing and input scanning. This examination allows the team to understand how the system's components function and respond under typical conditions. Automated testing helps quickly identify the system's potential vulnerabilities, and automated scanning, data injection, and fuzzing are effective ways to achieve a foundational level of security testing. The team uses the information gathered in this phase to conduct further manual testing

DISCOVERY & VULNERABILITY SCANNING

APPLICATION SCANNING	For identifying vulnerabilities and analyzing the security of web-based components of the product, the assessment team may use the following application scanners: •WebInspect • Burp Suite
FUZZ TESTING	The assessment team may conduct both automated and manual fuzz testing on various interfaces, focusing on all applicable entry points of the product, especially when it involves non-conventional, customized, or proprietary systems or protocols. Fuzz testing involves feeding invalid, unexpected, or random data to the inputs of the target system while monitoring it for crashes, failed assertions, or memory leaks. These entry points may include but are not limited to, application input fields, application protocols, network interfaces, and files.
THREAT IDENTIFICATION	The assessment team evaluates collected data to identify potential security or privacy risks through threat modeling, identifying attacker types, objectives, and attack methods. These threats are ranked by risk and used to develop attack plans for manual testing.

PHASE 3 :
MANUAL PRODUCT TESTING & CODE ANALYSIS

Automated scanning tools can save time on basic checks, but they shouldn't be a substitute for manual assessment. The team focuses on areas of concern, including those that pose potential security risks, during the manual assessment.

MANUAL TESTING & CODE ANALYSIS	
SCANNING VALIDATION	The team manually confirms every result from the automated scanners to eliminate any false alarms
CODE ANALYSIS	In addition to automated penetration testing, the assessment team conducts source code analysis (if provided) to verify penetration test findings and speed up exploit development by identifying specific logic flows and input requirements. This analysis leads to a more thorough review. The team identifies security controls and business logic to exploit, and examines the entire code for issues that may not be obvious from penetration testing or can be found more quickly through code review. The team searches for vulnerabilities in areas such as: • Architecture and business logic flaws • Authentication and authorization bypass • Insecure session management • Cryptographic weaknesses • Improper cryptographic module implementation • Client-side validation bypass • Manipulation of back-end services or calls • Leveraging file transfer capability • Inadequate input validation • Buffer overflow conditions • Potential manipulation of variables • Potential acceptance of external scripts or inputs • SQL injection • Command redirections • Problems with dynamic content creation • Accidental actions

CODE ANALYSIS (CONTINUED)	• Failure scenarios • Use of unsafe functions • Inadequate error management Depending on the system, the team may also reverse-engineer some components to gather more information for further testing.
SOFTWARE TESTING & ANALYSIS	The assessment team conducts expert-led penetration testing and analysis of the product using various tools such as network sniffers, attack proxies, file systems, process, and memory analysis tools, hardware and software debuggers, and custom-built attack tools. The team attempts to identify issues in areas such as: •Application Prioritization - determining the criticality of applications based on the handling of sensitive information such as PII, PHI, rights-restricted data, or other sensitive information, by analyzing factors like native language use, protocol and parser support, input validation, and encryption • Code Security - identifying coding and implementation issues such as buffer overflows, heap overflows, integer overflows, and off-by-one errors • Data Injection - injecting malicious data into the applications to alter the system's behavior or state • Data Interception - analyzing the communication mechanisms used by the product's subsystems to determine if messages at the hardware level can be intercepted • Data Replay - retransmitting data to bypass the security or application logic of specific product components, including hardware subsystems, firmware, application logic, or protocol parsers • Denial of Service - degrading service and rendering the product permanently or temporarily unavailable to legitimate users • Elevation of Privilege - taking steps that could ultimately allow an attacker to perform actions the product does not intend to permit, up to and including the arbitrary execution of program code on or within the target environment

MANUAL TESTING & CODE ANALYSIS

SOFTWARE TESTING & ANALYSIS (CONTINUED)	• Encryption Analysis - identifying supported encryption functionality and analyzing encrypted information that could lead to an attack against custom network protocols, network APIs, and applications or product firmware • Firmware Security - circumventing critical authentication or authorization functionality or the overall loading process to load unauthorized firmware, manipulate valid firmware, downgrade the product to an older firmware version, or otherwise modify firmware verification and loading behavior • Information Disclosure - intercepting, modifying, or deleting key information related to the product's security • Message Injection - injecting malicious data into the parsers to alter the parser's behavior or state • Message Manipulation - altering network messages intended for processing by the network protocols and parsers through manual testing and tools • Parser Security - reviewing the client's protocol parsers to identify if open source or commercial parsers have been used and analyzing whether known and unknown vulnerabilities have been mitigated • Protocol Enumeration - identifying custom protocols supported by the product and analyzing each protocol's intended use • Side Channel Leakage - transmitting sensitive information through a covert communication mechanism • Traffic Analysis - reviewing the traffic sent to and from the product's subsystems at the hardware level to determine whether sensitive data is transmitted Specific exploits are developed as needed and time permits, to demonstrate the vulnerabilities found during this phase of the assessment.

PHASE 4 : ANALYSIS AND REPORTING

Qualysec reports providing an executive summary of the assessment's goals, high-impact findings, and recommendations. Each finding includes a vulnerability definition, reproduction steps, business impact, and tailored recommendations with references. The team also builds a holistic view of the business risk for each finding.

TECHNICAL ANALYSIS ACTIVITIES	
LIKELIHOOD DETERMINATION	For each vulnerability, the assessment team evaluates the probability that it will be exploited based on the following criteria: • The motivation and ability of the potential attacker • The characteristics of the vulnerability • The presence and effectiveness of countermeasures
IMPACT ANALYSIS	For each vulnerability that could be exploited successfully, the assessment team assesses the impact on the organization and its customers in terms of confidentiality, integrity, and availability.
SEVERITY DETERMINATION	Qualysec assigns severity ratings using internal expertise and industry-standard methodologies such as OWASP and CVSS. Each finding's severity is evaluated separately, regardless of other findings. Vulnerabilities with higher severity ratings have a greater technical and business impact and require fewer dependencies on other issues.

PHASE 5 : REMEDIATION REVIEW

Additionally, if desired, the evaluation squad can repeat the examination and examination of discovered vulnerabilities after the customer confirms that the weaknesses have been remedied.