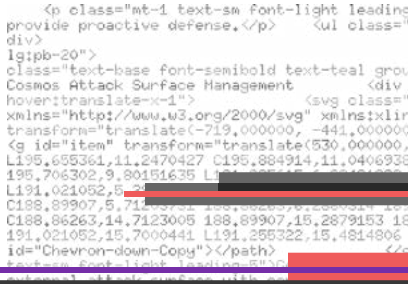
BISHOPFOX

METHODOLOGY

Bishop Fox's Hardware Penetration Testing methodology leverages cutting-edge hardware and software security assessment techniques to holistically assess products and their related infrastructure and system. Each test begins with the modeling of practical and theoretical threats against the system, including the consideration of product-specific factors such as the operating environment, users, and the sensitivity of data processed. The methodology outlined in this document provides a detailed look at the step-by-step process and delineation of responsibilities that are critical to accomplishing predetermined objectives.

# HARDWARE PENETRATION TESTING



# SUMMARY OF ENGAGEMENT

Before the engagement can begin, Bishop Fox's consultants will require the in-scope hardware, any detailed product documentation and access to the product deployment environment as well as credentials for all in-scope applications, and if possible relevant source code. In addition, Bishop Fox consultants will meet with all invested stakeholders to understand and confirm the objectives and scope of the engagement.

To kickoff the assessment, the team begins by leveraging all the pre-assessment information provided to build a model attack plan against the system, targeting areas that are likely to interest attackers. The team explores each area using attack techniques based on both past assessments and the latest security research. The assessment team then reviews the operation of the system under normal, non-assessment conditions through the automated testing and scanning of noted inputs allowing the team to observe how product components behave and react under normal use. The automated testing serves to rapidly map the product's attack surface and achieve a baseline level of security test coverage. The assessment team then leverages the information discovered in this phase during manual testing.

During the manual testing and validation phase, the team targets areas of specific interest likely to cause security concerns, identifying the specific security controls and business logic functionality to exploit. The team reviews the findings for issues that may not be obvious from penetration testing, including validating all automated scanning with context, code analysis and software testing and analysis.

Once weaknesses have been discovered, consultants document their findings including successful exploitation. Remediation steps applicable to each threat are outlined in detail and reviewed with stakeholders for feedback and clarification. Once feedback has been applied, reports are finalized and communicated to all parties. It is important to note that the primary outcome of this engagement is ensuring your security teams understand all findings and how they affect the product's organization and its customers.

# HIGH LEVEL PROCESS

1 2 3 4

### PRODUCT DOCUMENTATION PROVISIONS & ACCESS

- Receive product hardware, documentation, environment / credential access and source code.
- Work with the client to confirm objectives and scope of the engagement.

### MAP DEVICE ATTACK SURFACE & THREAT MODEL

- Build a Threat Model by identifying threats across the entire ecosystem of the device.
- The product's attack surface is mapped to understand the baseline security test coverage. The information is then used for manual testing.

### MANUAL PRODUCT TESTING

- Validate the mapped attack surface manually and perform multi-point analyses across the hardware and applications related to the device's ecosystem.

### ANALYSIS & REPORTING

- Determine critical and core vulnerabilities and determine remediation efforts required.
- Draft and review findings.
- Deliver three core components within their findings report: a Likelihood determination, Impact analysis, Severity determination, and if required perform a Remediation review.

# METHODOLOGY DETAILS

## PHASE 1: PRE-ASSESSMENT

The following assessment requirements must be met to ensure the timely and successful completion of the project.

### PRODUCT DOCUMENTATION

The assessment team requires detailed documentation, including but not limited to:

- Any available documentation related to the application
- A completed Application Assessment Scoping Survey (Optional and generally recommended for applications with complex environments or functionality.)

### ENVIRONMENT ACCESS

The assessment team may need access to the following resources related to the product deployment environment:

- Information on the underlying servers, operating systems, middleware, firmware, network or application access, and any third-party dependencies
- Encryption/decryption keys or passphrases required to access the hardware or software

### CREDENTIALS

The assessment team may need access to the following application resources, including but not limited to:

- A minimum of two sets of credentials for each role
- Any information required to use the product (e.g., password reset information, security-question answers, and secure tokens)

### SOURCE CODE

The assessment team may need access to product source code, including:

- 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 product hardware, including but not limited to:

- Production hardware, including any necessary support hardware
- Development or debug versions of multiple units of the hardware
- Programming or on-chip debugging (OCD) hardware if needed to load the firmware
- Any tools or test harnesses

### DUE CARE

Throughout the assessment, Bishop Fox makes an effort to minimize disruptions to network availability, particularly when performing any automated scanning, manual validation, or penetration testing. Prior to testing, the assessment team will discuss risks to environmental stability with the client and identify the escalation path if any disruptions are observed.

### AUTHORITY

If any portion of the product or related resources is hosted on a third-party system, a consent to test must be obtained prior to the start of fieldwork.

| | BISHOP FOX | CLIENT |
|---|---|---|
| INCLUDE SECURITY TEAM AND BUSINESS STAKEHOLDERS | ✓ | ✓ |
| PROVISION ALL PRODUCT DEPLOYMENT DOCUMENTATION | | ✓ |
| PROVISION CREDENTIALS AND ENVIRONMENT ACCESS | | ✓ |
| PROVIDE HARDWARE | | ✓ |

## **PHASE 2:** INFORMATION GATHERING & AUTOMATED TESTING

The assessment team begins this phase by reviewing the operation of the system under normal, non-assessment conditions through the automated testing and scanning of noted inputs. This review allows the team to observe how product components behave and react under normal use. The automated testing serves to rapidly map the product's attack surface, and automated application scanning, data injection, and fuzzing are all powerful methods for effectively achieving a baseline level of security test coverage. The assessment team then leverages the information discovered in this phase during manual testing.

### APPLICATION SCANNING

For any of the product's web application or web service components, the following application security scanners may be used to detect vulnerabilities and map the surface of the application:

- HP WebInspect
- Burp Suite

### FUZZ TESTING

Automated and manual fuzzing may be performed on a per-interface basis for all relevant product entry points, especially when non-standard, custom, or proprietary systems or protocols are used. Fuzz testing provides invalid, unexpected, or random data to the inputs of a target while it is being monitored to detect crashes, identify failed assertions, or locate memory leaks. Entry points may include but are not limited to application user input fields, application protocols, network interfaces, and files.

### THREAT IDENTIFICATION

After reviewing the system's intended use, behaviors, and operating conditions, the assessment team reviews the information collected to identify the areas most likely to generate security or privacy issues. This threat modeling identifies the type of threat actor, the threat actor's likely goals against the product, and the most efficient means of attack to accomplish these goals. These threats are ranked by the likely risk posed to the product owner and then used to build the attack plans employed in the manual testing phase.

| | BISHOP FOX | CLIENT |
|---|---|---|
| PERFORM APPLICATION SCANNING | ✓ | |
| PERFORM MANUAL AND AUTOMATED FUZZ TESTING | ✓ | |
| REVIEW THE PRODUCTS SYSTEM FOR NORMAL OPERATING CONDITIONS | ✓ | |
| CREATE A THREAT MODEL | ✓ | |
| BUILD AN ATTACK PLAN FOR THE MANUAL TESTING PHASE | ✓ | |

# **PHASE 3:** MANUAL PRODUCT TESTING & CODE ANALYSIS

While automated scanning tools can significantly reduce the amount of time required to perform basic application checks, they should not replace a manual assessment. During the manual assessment, the team targets areas of specific interest, including areas likely to cause security concerns.

## SCANNING VALIDATION

The assessment team manually validates every finding from the automated scanners to eliminate any false positives.

## CODE ANALYSIS

Along with automated and penetration testing, the assessment team performs source-code analysis (if source code is provided). By analyzing the source code, the team can verify penetration-test findings and shorten the exploit development process by determining exact logic flows and input requirements. This analysis can result in a significantly more effective overall review. During this process, the assessment team identifies the specific security controls and business logic functionality to exploit. The team then reviews the entire code for issues that may not be obvious from penetration testing or may be found more rapidly by code review. Throughout, the team attempts to identify product vulnerabilities in the following areas:

- Architecture and business logic flaws
- Authentication and authorization bypass
- Insecure session management
- Cryptographic weaknesses
- Improper implementation of cryptographic modules
- 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
- Dynamic content creation issues
- Unintended operation
- Failure conditions
- Use of insecure functions
- Improper error handling
- Potential manipulation of variables
- Potential acceptance of external scripts or inputs
- SQL injection
- Command redirections
- Dynamic content creation issues
- Unintended operation
- Failure conditions
- Use of insecure functions
- Improper error handling

Depending on the nature of the system, the team may also attempt to reverse-engineer select components of the system to provide additional information for further attacks.

## SOFTWARE TESTING & ANALYSIS

The assessment team performs expert-guided penetration testing and analysis of the product using a variety of tools, including network sniffers; attack proxies; file system, process, and memory analysis tools; hardware and software debuggers; and custom-built attack tools. The assessment team attempts to explore and identify issues in the following areas:

- **Application Prioritization** — identifying high-criticality applications running on the product, based on personally identifiable information (PII), protected health information (PHI), rights-restricted data, or other sensitive information
- **Code Security** — identifying insecure 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, resulting in the alteration of the system's behavior or state
- **Data Interception** — analyzing the communication mechanisms used by the product's subsystems to determine whether messages at the hardware level can be read
- **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
- **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** — bypassing 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, resulting in the alteration of the parser's behavior or state
- **Message Manipulation** — using manual testing and tools to alter network messages intended for processing by the network protocols and parsers
- **Parser Security** — reviewing the client's protocol parsers to identify if open source or commercial parsers have been used, and then 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 constructed, as required and as time allows, that demonstrate the vulnerabilities found during this phase of the assessment.

| | BISHOP FOX | CLIENT |
|---|---|---|
| MANUALLY VALIDATE AUTOMATED RESULTS | ✓ | |
| ELIMINATE FALSE POSITIVES FROM AUTOMATED RESULTS | ✓ | |
| PERFORM SOURCE CODE ANALYSIS (IF PROVIDED) | ✓ | |
| REVERSE-ENGINEER SYSTEM COMPONENTS FOR ADDITIONAL ATTACK CONTEXT | ✓ | |
| ANALYZE SOFTWARE, TEST, AND CONSTRUCT EXPLOITATION | ✓ | |
| EXPLOIT VULNERABILITIES AND WEAKNESSES | ✓ | |

## **PHASE 4:** ANALYSIS & REPORTING

Bishop Fox reports contain an executive-level summary of the engagement, which includes the assessment's goals, a synthesis of the highest-impact findings, and high-level recommendations. Within each finding, a vulnerability definition is given along with detailed reproduction steps, a description of the business impact, and tailored recommendations with references.

For each finding, the assessment team builds a holistic view of the business risk it represents by performing the following activities.

### LIKELIHOOD DETERMINATION

For each vulnerability, the assessment team determines the likelihood that it will be exploited based on the following factors:

- Threat-source motivation and capability
- Nature of the vulnerability
- Existence and effectiveness of controls

### IMPACT ANALYSIS

For each vulnerability, the assessment team analyzes and determines the impact of successful exploitation as it affects the organization and its customers in the areas of confidentiality, integrity, and availability.

### SEVERITY DETERMINATION

Bishop Fox determines severity ratings using in-house expertise and industry-standard rating methodologies such as the Open Web Application Security Project (OWASP) and the Common Vulnerability Scoring System (CVSS) to evaluate the likelihood and impact of exploitation. The team weighs those factors to classify the overall severity as critical, high, medium, or low. The severity of each finding is determined independently of the severity of other findings.

| | BISHOP FOX | CLIENT |
|---|---|---|
| LIKELIHOOD DETERMINATION | ✓ | |
| IMPACT ANALYSIS | ✓ | ✓ |
| SEVERITY DETERMINATION | ✓ | |

## **PHASE 5:** REMEDIATION REVIEW (OPTIONAL)

Optionally, the assessment team re-performs scanning and testing of the identified vulnerabilities after the client indicates that the vulnerabilities have been addressed.

# APPENDIX

## BISHOP FOX ACT DELINEATION OF RESPONSIBILITIES

| | BISHOP FOX | CLIENT |
|---|---|---|
| **PHASE 1: PRE-ASSESSMENT REQUIREMENTS** | | |
| IDENTIFY AND MEET WITH SECURITY TEAM AND BUSINESS STAKEHOLDERS | ✓ | ✓ |
| PROVISION ALL PRODUCT DEPLOYMENT DOCUMENTATION | | ✓ |
| PROVISION CREDENTIALS AND ENVIRONMENT ACCESS | | ✓ |
| PROVIDE HARDWARE | | ✓ |
| PROVIDE SOURCE CODE (IF AVAILABLE) | | ✓ |
| **PHASE 2: INFORMATION GATHERING & AUTOMATED TESTING** | | |
| PERFORM APPLICATION SCANNING | ✓ | |
| PERFORM MANUAL AND AUTOMATED FUZZ TESTING | ✓ | |
| REVIEW THE PRODUCTS SYSTEM FOR NORMAL OPERATING CONDITIONS | ✓ | |
| CREATE A THREAT MODEL | ✓ | |
| BUILD AN ATTACK PLAN FOR THE MANUAL TESTING PHASE | ✓ | |
| **PHASE 3: MANUAL PRODUCT TESTING & CODE ANALYSIS** | | |
| MANUALLY VALIDATE AUTOMATED RESULTS | ✓ | |
| ELIMINATE FALSE POSITIVES FROM AUTOMATED RESULTS | ✓ | |
| PERFORM SOURCE CODE ANALYSIS (IF PROVIDED) | ✓ | |
| REVERSE-ENGINEER SYSTEM COMPONENTS FOR ADDITIONAL ATTACK CONTEXT | ✓ | |
| ANALYZE SOFTWARE, TEST, AND CONSTRUCT EXPLOITATION | ✓ | |
| EXPLOIT VULNERABILITIES AND WEAKNESSES | ✓ | |
| **PHASE 4: ANALYSIS & REPORTING** | | |
| LIKELIHOOD DETERMINATION | ✓ | |
| IMPACT ANALYSIS | ✓ | ✓ |
| SEVERITY DETERMINATION | ✓ | |