WHITE PAPER

THE LARES APPSEC ASSESSMENT METHODOLOGY

APPLICATION SECURITY ASSESSMENT

The objective of an application assessment is to determine the application's overall security and the communication between the application's different components and back-end systems. By performing an application assessment, Lares can ensure the appropriate controls are implemented within the application to confidentiality, integrity, and availability of the information

ABOUT LARES

Lares is a security consulting firm that helps companies secure electronic, physical, intellectual, and financial assets through a unique blend of assessment, testing, and coaching since 2008.

For more information, visit lares.com, contact us at (720) 600-0329, or follow Lares on Twitter @Lares_.

www.lares.com

CONTACT INFORMATION

1580 N. Logan St. Ste 660, PMB 79199
Denver, CO 80203, USA

+1 (720) 600-0329
sales@lares.com

Executive Summary

Application Assessments are designed to assess the development efforts of commercial and/or third-party applications. In contrast to network-based assessments, application assessments identify design flaws and make recommendations to improve security at the development level. Additionally, performing application assessments provides a means to ensure and reduce risk early in the development process.

The objective of an application assessment is to determine the application's overall security and the communication between the application's different components and back-end systems. By performing an application assessment, Lares can ensure the appropriate controls are implemented within the application to confidentiality, integrity, and availability of the information.

Methodology Overview

- Review and Analysis
- Vulnerability Exploitation
- Documentation and Reporting

Assessment Type Determination

Having access to the source code provides a more sound approach to assessing the security of an application during the development phase. When the source code is available, a Gray Box assessment is performed with a limited amount of Black Box testing. If the source code is not available, a full Black Box assessment is performed.

If code is available for the application a combination of gray box, black box, and network testing will be conducted. The specific actions of each assessment type are outlined in the following sections.

Grey Box

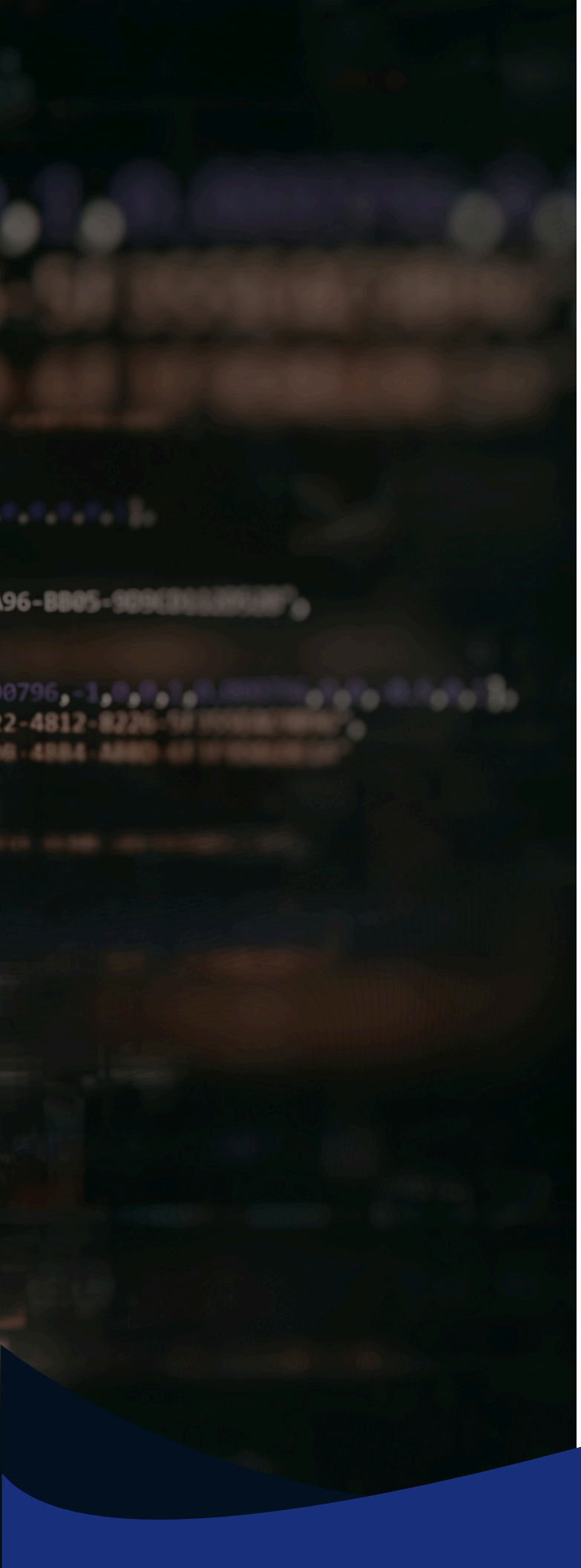
A Grey Box review is conducted when the source is available for the application in scope. Gray box testing provides more visibility of the application and back-end communication to systems that may not visible during a standard black box assessment. As part of a gray box review, the following activities are conducted:

- Source Code Reviews – third-party applications and manual reviews are utilized to identify problems in code development before applications move further into the development lifecycle. The identified and remediated vulnerability can then be validated during the Black box assessment.
- Application scan – as part of the gray box review, an automated application scan will be conducted by Lares. One example of an automated tool selected for application scans is HP (SPI Dynamics) WebInspect.
- Validation – is the process of confirming identified vulnerabilities during the automated source code and application level scans. During this phase, exploitation and focused attacks occur at specific portions of the application to review the output and final state of the application in scope and affected back-end systems.
- Manual Test - manual testing is always required even after automated scans occur. Automated scan tools often find the “low-hanging fruit” and indicate areas where more focus should be placed. Manual testing is a mixture of automated findings reviews and hands-on testing to determine the outcome of potential attacks.

Black Box

A Black Box review is conducted when no source code is available for the application in scope. Often Black Box reviews are performed when there is limited assessment time, or the application is already in production. Black Box reviews are also useful for retesting over a longer period, such as quarterly assessments. As part of the Black Box review, the following activities are conducted:

- Application scan – as part of the gray box review, an automated application scan will be conducted by Lares. One example of an automated tool selected for application scans is HP (SPI Dynamics) WebInspect.
  Validation – confirming identified vulnerabilities during the tools' automated source code and application-level scans. During this phase, exploitation and focused attacks occur at specific portions of the application to review the output and final state of the application in scope and affected back-end systems.
- Manual Test – manual testing is always required even after automated scans occur. Automated scan tools often find the “low-hanging fruit” and indicate areas where more focus should be placed. Manual testing is a mixture of automated report reviews and hands-on testing to determine the outcome of potential attacks.
- Reporting – once all automated and manual testing is complete, reports are prepared to deliver to the application owner(s). Reports contain details regarding each finding with evidence obtained during validation. Recommendations for remediation are also included.



Application Assessment Process

The following phases are conducted during a typical web application assessment.

Review and Analysis

Web Content Review

- Perform a web crawl over the entire website (or all accessible portions), and download it to the local machine.
- Manually review the website from the web user's perspective. Specifically, identify:
  - Company information (Addresses, telephone numbers, company subsidiaries);
  - Contact information (Employee names, email addresses, and naming conventions);
  - Links to other websites, companies, or vendors.
- Determine the web server and operating system version.
- Search for a list of known files to help enumerate the installed web applications and identify their software versions.
- Send a raw request to the server and examine the results.
- Scan the service ports on the machine for other web server ports, and determine the operating system (optional).

Web Code Review

This step of the process involves reviewing the code that is accessible in text format to individuals accessing the site. This code includes the web pages - HTML, JavaScript, or VBScript code, and is visible in plaintext whether the connection is plain HTTP or secure HTTPS.

Collect and identify website areas that could be abused to gain unauthorized access or force the application to behave in an undesirable manner. The following steps should be customized (as appropriate) for each website:

- View the source code of all pages and frames within the scope of the engagement. Examine comments embedded within the web pages, the JavaScript / VBScript files, and the code.
- Search the web pages for hidden fields. Some sites rely on ‘hidden’ form fields to pass potentially sensitive information from one web page to the next in a sequential process. These form fields are not actually ‘hidden’ but are available to anyone who can view the HTML source of the web page. These fields can also be changed easily; hence, malicious data can be interjected into the application processing.
- Search the web pages for the use of forms and for posting these forms via HTTP GET or POST methods to relay data back to the Web Server.
- Look for places where session information is transmitted to other sites, such as a merchant or shopping cart site.
- Attempt to identify the website infrastructure from the file suffixes (e.g., PHP uses ".php", ASP uses ".asp", etc.) or by observing the error messages.
- Issue a request for a non-existent page (such as *http://www.abccompany.com/abc_xyz*), and check for important information in the error messages.

Vulnerability Identification

Review the software platforms used in the applications and compile a list of known vulnerabilities associated with these platforms/software packages.

Identify vulnerabilities based on category, type, and threat level.

A sampling of information on specific web application vulnerability categories covered within the scope of testing can be found here:

- https://owasp.org/Top10/
- http://projects.webappsec.org/w/page/13246974/Threat%20Classification%20Reference%20Grid

Security Issue Identification

Analyzing the data and information collected during the previous steps is important to make informed decisions. This data can help identify patterns and trends that may not be immediately apparent.

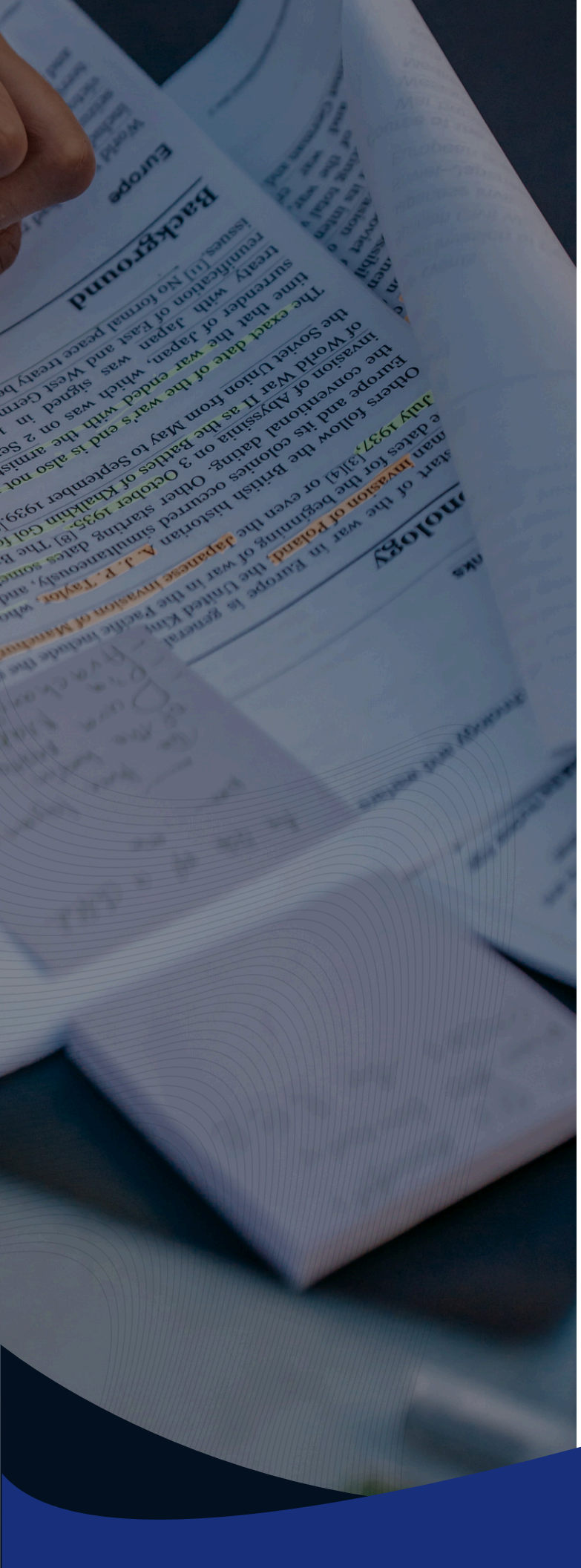
By analyzing this data thoroughly, you can gain valuable insights that can improve your decision-making abilities.

Vulnerability Exploitation

This phase aims to test, validate, and exploit potential vulnerabilities inherent in the commercial off-the-shelf software used in the applications being reviewed or in the custom-developed applications.

From the data produced in the previous phase, analyze the application architecture and HTML source code, including the use of forms, URLs, hidden form fields, JavaScript, etc., publicly accessible to any web surfer. Use this information to modify some of these fields, note the application response and behavior, and then exploit these vulnerabilities to modify the application's intended behavior. This general approach includes the following:

- Modify any/all parameters in any of the URLs. For numeric parameters, attempt to increment/decrement the numbers or delete the parameter altogether. For alphanumeric parameters, attempt to append or replace contents with "*", "..", "/", ";", "|" or ‘garbage’ such as "xxx."
- Many sites use cookies to maintain session information. Configure the browser to "Warn me before accepting cookies" to see what the cookies are used for.
- Change the permanent or date-dependent cookies (found in the Internet temporary files directories).
- Change the temporary cookies.
- Identify login names, session/user IDs, signatures, and attempt to add/modify/delete them.
- Send cookies that are too large or two cookies under the same name.
- Access some pages without the ASPSESSION_ID cookie - some ASP pages do not assure that there is a session active.
- Guess or manipulate the session ID to access another user’s session.
- Implement buffer overflow techniques with web forms by passing a large data block to a form handler on the server.
- Manipulate the hidden fields, and submit forms using these manipulated fields.
- Gain access to the directory structure by finding a directory traversal-enabled area of the website.
- Circumnavigate server-side input validation functions by programmatically passing random information to form posts or URLs.
- Cross Site Scripting (XSS) attack to determine whether code can be inserted.
- SQL injection (SQLi) on form fields to determine if user input is correctly validated and if database enumeration is possible.



Documentation and Reporting

This phase involves documenting the results of the previous phases and making recommendations to resolve any issues discovered.

After reviewing the results from the previous phases, document the results in a summary report, and develop recommendations for improvement in the application architecture, design, or code. Results are commonly reported at two levels:

- Tailored to executive management -- summarizing the issues and related risks and providing examples of the vulnerabilities.
- Tailored for application developers and security specialists -- itemizing the vulnerabilities, risks, exploitations, and recommendations.

The report is written following ISACA IS Auditing Standards, including:

- Scope definition
- Objectives
- Period of work performed
- Nature, timing, and extent of the assessment performed
- Findings and conclusions as to the effectiveness of controls
- Recommendations on remediating the vulnerabilities identified

Conclusion

Application assessments are an important step in ensuring the security of your applications. By identifying design flaws and security issues early in the development process, application assessments can help reduce risk. At Lares, we have extensive experience performing application assessments for various organizations. We use various assessment methods to identify vulnerabilities and security issues, including black box, grey box, and web content reviews. Our team of experts will work with you to scope your next application security assessment and ensure that your applications are as secure as possible.

If you want to learn more about application security assessments or how our experts can help scope and execute an effective test for your business, please contact us. We would be happy to discuss the many ways in which we can help keep your organization safe. Thank you for reading!

ABOUT LARES

Lares is a security consulting firm that helps companies secure electronic, physical, intellectual, and financial assets through a unique blend of assessment, testing, and coaching since 2008.

For more information, visit lares.com, contact us at (720) 600-0329, or follow Lares on Twitter @Lares_.

www.lares.com

CONTACT INFORMATION

1580 N. Logan St. Ste 660, PMB 79199
Denver, CO 80203, USA

+1 (720) 600-0329
sales@lares.com