lemlock

Learn more about Lemlock Scans

www.lemlock.com

Table of Contents

Aim of the document

Choosing the right type of security scan can be problematic, but it doesn't have to be. The document you are currently reading has been prepared so that you can familiarize yourself with the available types of Lemlock scans. We want your final decision to be fully rational and adequate for the business needs of your application.

Learn about the types of Lemlock scans and areas that are verified in terms of your web solution.

PASSIVE SCAN

ACTIVE SCAN
BASIC

ACTIVE SCAN
FULL

1 What are the differences between the types of Lemlock scans?

The types of scans differ mainly in the **areas of scanning** and the **depth of scans**. A higher type is always associated with the descent of the scanner into the lower parts of the application, which results in more accurate verification of the components of the web solution.

CHECK WHAT SPECIFIC SECURITY ACTIVITIES ARE UNDERTAKEN AS PART OF EACH SCAN:

TYPE OF ACTION	PASSIVE	ACTIVE BASIC	ACTIVE FULL
Application errors			
CSRF countermeasures			
Cache control			
Charset mismatch			
Content Security Policy headers			
Content-Type headers			
XSS protection headers			
Frame-Options headers			
Cookie security (httpOnly, Secure)			
Cross domain script inclusion			
Debugging information disclosure			
Insecure authentication			
Insecure JSF ViewState			
Mixed content (HTTPS)			
Private IP information disclosure			
Insecure Session ID			
Cross-site scripting (reflected)	x		
Cross-site scripting (stored)	x		
CRLF injection	x		
Directory browsing	x		
External redirect	x		
Parameter tampering	x		
Path traversal	x		
Source code disclosure	x		
SQL injection	x		
Buffer overflow	x	x	
Code injection	x	x	
Server-side include	x	x	
Command injection	x	x	
Format string error	x	x	
Remote file include	x	x	

2 What kind of activities will be carried out as part of the scan?

Below you will find an alphabetical list containing the **types of activities including their detailed and technical descriptions** within the available types of Lemlock scans.
Familiarize yourself with it to be sure what types of activities will be carried out in your application and what hacker attacks you can avoid after implementing the technical recommendations from the obtained final report:

TYPE OF ACTION	DESCRIPTION OF ACTION
Application errors	Detecting error notifications in backend solutions such as databases or HTTP servers. Misconfigured solutions can reveal the database structure or server version and facilitate hackers to break in.
Buffer overflow	Testing the vulnerability of applications for buffer overflow attacks. The buffer overflow means that the application has been forced to accept more input than the programmer had anticipated (injecting a string containing redundant data). Modern programming languages and operating systems have methods of protection against such attacks, but they still happen and are difficult to detect. The consequence of such an attack is numerous disruptions in the proper operation of the application or - in some situations - the execution of the code contained in the aforementioned redundant data.
Cache control	Detecting of incorrect cache control directives that the browser uses to cache client data. Some data should not be cached at all due to confidentiality, e.g. session tokens. What's more, wrong directives also cause problems with website performance.
Charset mismatch	Detecting of different character sets between HTTP requests and HTML content of the page. Incorrectly configured allowable character sets can result that browser security filters are not working properly.
Code injection	Testing code injections through susceptible web application parameters. If the injected snippet of code executes and its result is found in the server's response, then it will suggest a serious vulnerability that allows you to change the application's source code. The result of code injection may be manipulating its operation and forcing the server to disclose confidential information or to communicate with another server. In the event of an effective attack, the hacker can gain full access to the operating system.
Command injection	An attempt by the scanner to inject operating system shell commands (such as Linux or Windows) into the parameters adopted by the web application. The scanner has many test payloads that check the operation of various commands in the operating system. If such an attack is successful, the hacker can gain full access to the operating system. In addition, if the server has administrative privileges, the hacker will also get them. However, if the server does not have administrative privileges, it is often possible to escalate the privileges from the obtained shell with server privileges.
Content Security Policy headers	Analysis of problems in Content Security Policy (CSP) headers. Properly configured CSP makes it very difficult for hackers to carry out attacks based on JS code.
Content-Type headers	Analysis of problems in Content-Type headers. The missing or incorrect Content-Type header forces the browser to guess the content type, which can sometimes result in the execution of malicious JS code embedded, e.g. in an image.

TYPE OF ACTION	DESCRIPTION OF ACTION
Cookie security (httpOnly, Secure)	Checking the correctness of cookie directives. The HttpOnly directive prevents scripts - including malicious ones - from reading cookies, and the Secure directive only allows cookies to be sent via a secure HTTPS channel.
CRLF injection	The scanner attempts to inject CRLF control characters, i.e. the end of line characters \r and \n. A smooth injection allows hackers to take control of server responses. As a result, the browser that encounters such characters will stop processing the server's response, which may be associated with undesirable effects and effectively affecting the appearance of the website. An example of such an attack could be the "deface" of the site, in other words online vandalism, consisting of manipulating the content of the site. The content of the page is changed and the end user sees this content in their browser, assuming that this was the intention of the authors. In addition, hackers in this way perform cache poisoning attacks and transmit cache-substituted content to clients. This vulnerability also enables XSS reflected attacks, which are described in this table.
Cross domain script inclusion	Source validation of embedded scripts. Analysis of the source of scripts embedded in the site can sometimes detect injections of malicious code.
Cross-site scripting (reflected)	The scanner examines the possibility of injecting the interpreted string (as script code, usually JS) into the parameter present on the page, the value of which is then rendered in the client's browser. If such an attack succeeds, it will be possible to embed malicious code in the user's browser. Malicious code can be provided, e.g. in the form of a link to the application, with encrypted and malicious payload so that the user will not pay attention to the malicious link. Part of the application domain and HTTPS will remain intact and the user will be deceived by using his/her trust in the site.
Cross-site scripting (stored)	A similar attack to the Cross-site scripting (reflected) attack described in this table, except that the injection of the interpreted string is tested in the context of the database. The scanner tries to put its own piece of code in the database, which will then be used as part of the website, e.g. in the form of a comment stored in the database. It is worth noting that the scanner only injects strings to prove the presence of vulnerabilities, but without causing any harmful actions. If a hacker finds a Cross-site scripting (stored) vulnerability, in the next step it will be able to embed the malicious code on the page, which will be executed in the browsers of every user who visits the page. Thus, the attack is more serious than Cross-site scripting (reflected) due to the ease of providing payload to the user - you do not need to convince him/her to click on the link.
CSRF countermeasures	Detecting the presence of tokens against Cross-site request forgery attacks. The lack of tokens allows hackers to create effective phishing attacks with links to the site. By clicking on these links as a logged-in user, you can unknowingly carry out an action on your own account, e.g. in the form of its removal or extraction of confidential information.
Debugging information disclosure	Detecting whether the application contains debugging code that can expose sensitive information to untrusted parties. When such situation is detected, the application should be rebuilt in release mode to remove sensitive information which increases stability and safety.
Directory browsing	The scanner analyzes website subpages in search of listed directories. There are cases where administrators misconfigure directory permissions within the server. This results in unauthorized access to a directory that only the operating system should have access to, not the client with the browser. The consequences are leaks of sensitive data or information about the server configuration, greatly facilitating further attacks. Usually, these are backup files containing all information from the database or configuration files including, e.g. access data.
External redirect	Using various techniques to redirect the user to the URL selected by the scanner. Effective redirection may cause the particular user who browses the website can be directed to a website belonging to cybercriminals, which facilitates a phishing attack. In another variant of the attack, hackers can simply cause damage to the site by placing malicious redirects to pages with inappropriate content.

TYPE OF ACTION	DESCRIPTION OF ACTION
Format string error	The scanner actively tries to cause an error in the application and thus recognize potential problems in the application's string formating The scan is focused on C-type compiled languages. It differs from passive searches by actively trying to cause these errors, which improves scan efficiency.
Frame-Options headers	Frame-Options headers control whether a page can be embedded in a frame on other pages. Allowing embedding of the page results in dangerous Clickjacking attacks. The attack consists of the user making clicks in an online space that looks just like the visited website. However, in reality, there is the interaction with malicious, transparent content substituted by cybercriminals.
Insecure authentication	Detecting a situation where HTTP credentials are sent over an explicit channel (without HTTPS). This situation allows cybercriminals to eavesdrop on transmitted access data and use it.
Insecure JSF ViewState	Detecting of cryptographically unsecured ViewState fields in which applications often store confidential information. A hacker who has obtained such information may be able to take over the victim's session and log into account.
Insecure Session ID	Detecting of session tokens sent in a non-secure manner, e.g. in a URL that is not protected by HTTPS. It is easier for cybercriminals to eavesdrop and consequently use such tokens to log in to someone else's account.
Mixed content (HTTPS)	Detecting a situation in which part of the website supports secure HTTPS, and the next part only HTTP. It is recommended to send the entire website content using HTTPS.
Parameter tampering	Attempts to inject parameters causing errors in the web application. The scan includes other injection techniques that are focused on server software (Apache, IIS). A successful injection results in the disclosure of internal information about software versions.
Path traversal	The scanner tries to access operating system files that are outside the directory accessible to the server. Sometimes it is possible to "climb up" the directory tree using special payloads with relative or absolute paths that point to confidential files, e.g. /etc/passwd. If you can read confidential operating system files, you can get key information in a further attack, such as a list of system users or configuration files containing access information.
Private IP information disclosure	Detecting of private IP addresses in server responses, which indicates incorrect configuration. An attacker who has information about internal servers has simplified all actions connected with more advanced attacks such as Server-Side Request Forgery.
Remote file include	The scanner examines the possibility of embedding external source code in the application code in backend terms. If you can force an application to attach a remote code, you can also force it to execute any kind of code. The attacker, by gaining the ability to execute the code of the choice on the server, in the next step is able to take full control over the server and fully violate the confidentiality, integrity, and availability of information.
Server-side include	The scanner is testing the possibility of injecting strings into an application that may force the server to return certain data defined by the application server directives. The directive can also execute the operating system shell command. Unlike other techniques, SSI works by manipulating the behavior of the server itself on which the application operates, and not the source code of the application. Consequently, confidential information is disclosed, or the hacker gets administrative privileges.
Source code disclosure	Searching for directories available for the application server written in Java, in which the compiled code in binary form can be stored. Code in this form hackers can very quickly lead to a readable form for them by using reverse engineering. An attacker who gains access to the application code has full information about its operation. In addition, it happens that developers save confidential data in the application source code, e.g. API keys or access data to other servers. They do this with the misconception that source code in compiled form is unreadable and additionally protected by server directory permissions.

TYPE OF ACTION	DESCRIPTION OF ACTION
SQL injection	Detection of SQL injections into the database made in an unauthenticated manner. If the application does not effectively filter the parameters from which SQL queries are created, this may result in interaction with the database. A successful SQL injection attack is one of the most serious attacks because databases usually store confidential information. In addition to data leakage, there is often the possibility of data modification or even escalation to the operating system shell.
XSS protection headers	Analysis of the correctness of protection headers against Cross-site scripting (XSS) attacks. A correctly configured header supports protection against XSS attacks by blocking the page from displaying when a malicious script is detected.

3 Is the scan suitable for my application?

Deciding to perform a scan and make the final decision on the choice of its specific type does not have to involve many hours of research. We decided to facilitate this process for you and prepare a short checkpoint checklist.

Give honest answers to the following points to find out if your application requires security scans and other actions to restore its high level of protection.

CHECKPOINTS	ANSWER	
Were the architecture and infrastructure of your application created taking into account security standards, e.g. OWASP, ASVS?	YES	NO
Have you tested the security of your application over the past year?	YES	NO
Has your application been verified in terms of resistance to vulnerabilities from the OWASP TOP10 ranking?	YES	NO
Does your application meet current and global security requirements of digital solutions, e.g. ISO 2700x or GDPR?	YES	NO
Have you ever tested your application using the "whitebox", "greybox" and "blackbox" methods?	YES	NO
Do you have a Security Officer in your team or another person who deals with application security?	YES	NO
Do you have a Personal Data Administrator in your team or another person who deals with the protection of collected and processed data?	YES	NO
Does your application not allow users to enter parameters into it?	YES	NO
Is your application unlisted?	YES	NO
Does your application not allow processing of confidential data?	YES	NO
Are you able to provide evidence of the appropriate application structure and effectiveness of the security mechanisms used (e.g. encryption mechanisms)?	YES	NO

How to interpret the obtained result?

More NO answers - There is a high risk of attack. It is necessary to verify the security of the application, and the scan is the desired beginning of the entire process.
More YES answers - Your application can be considered safe as long as the verification activities have been carried out correctly and the recommendations have been implemented immediately.

4 Is it worth getting support from a cybersecurity partner?

Lemlock was created in response to the increased demand for proven and cybersecured applications, both in technical and legal terms. Our main goal is not only to carry out activities in the form of analyzes and security audits but above all to support customers in implementing verified security measures to the company, as well as to the offered digital products or services.

Lemlock values

In our daily activities we are guided by four values, which are:

CLIENT-CENTRIC APPROACH

we provide comprehensive technical and legal cybersecurity support

RESPONSIBILITY

we have a sense of responsibility, which results in proper protection of those who have trusted us

RESPECT AND EQUALITY

we greatly respect our clients, colleagues, and partners

CONSTANT IMPROVEMENT

we are constantly learning about evolving cyber threats to be always one step ahead of cybercriminals

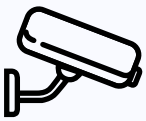
5 The comprehensive approach to cybersecurity

The security of IT solutions in each case is related to **technical** and **legal** areas. See our detailed offer to comprehensively approach the issue of security in your business.

Technical cybersecurity of application

Application Security Audits & Penetration Tests

Application GDPR Compliant Audits



Continuous Pentesting

Legal cybersecurity of application

Risk management & security policy

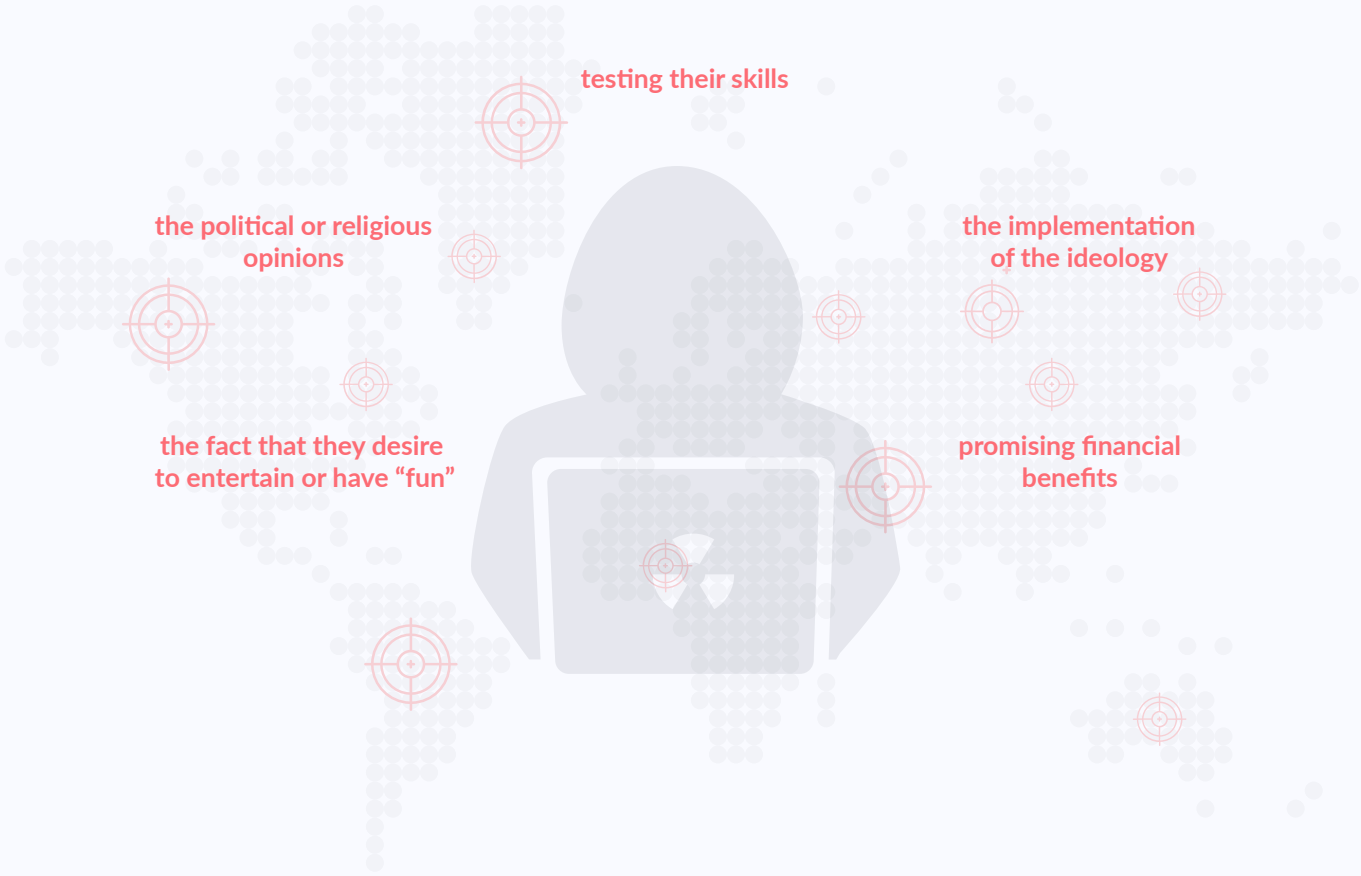
Incident handling and post-breach analysis

Staff training

FIND OUT MORE

Why are cyber attacks so dangerous to business?

Hacker motivations can have different grounds, most often they decide to launch an attack due to:



Carrying out hacker attacks on a company and making dangerous modifications to its infrastructure or data theft is one of the everyday phenomena. **It does not matter to the hacker what size company you represent or which industry your business belongs to** - any company with attractive resources or valuable information can become the object of a cyberattack.

The effects of a successful hacker attack are hard to be underestimated. Loss of key data, the devastation of IT systems or application blocking cause not only damage to the company’s image or reputation fall but also huge financial losses.

What are the effects of improper security of the application and digital assets?

loss of data enabling business operations

blockage of the organization's activity and its revenue

loss of reputation

destruction of IT systems

financial penalties up to 20 million euros or 4% of the annual global turnover for non-compliance with the GDPR guidelines

The security of applications, systems, and data is a continuous process that involves not only the implementation of mandatory standards or security measures. It is a process that is based primarily on continuous monitoring and testing.

Cybersecurity leaves no room for compromise.
Check regularly that your application or other systems are not creating opportunities for hackers.

How can I get more information?

Do you still have a dilemma, which Lemlock scan type will be best for your application?

Or maybe you need additional information on ongoing security activities?

Get dedicated support and contact our specialist:

Marcin Michalski
Business Development Partner

+48 71 700 03 01
marcin.michalski@lemlock.com
LinkedIn

SCHEDULE A MEETING