

Cryptosense Analyzer

Cryptosense Analyzer helps developers, application security teams and pen testers quickly identify hard to find cryptographic security flaws in applications.

The screenshot displays the Cryptosense Analyzer interface. At the top, it shows the 'Analyzer' logo and 'Demo Report' with options to rename, share, print, or export as CSV. The user profile is 'graham (logout)'. Below the header is a navigation bar with tabs for 'Summary', 'Algorithm (2)', 'Use of Crypto (3)', 'Key Length (1)', and 'Key Management (1)'. A 'Stack trace filter' input field and 'Go'/'Reset' buttons are also present.

The main content area is divided into two columns. The left column shows a list of 'Failed (7 rules)' findings:

- 2. Symmetric key too short**: The key is too short to provide adequate resistance to brute force attacks.
- 7. Initialisation vector 0 in CBC mode**: The initialisation vector (IV) should be a random value for this encryption mode.
- 21. Storage of keys in a Sun JKS key store**: This is a legacy keystore type with multiple weaknesses.
- 9. Use of PKCS#1v1.5 Encryption**: This padding mode is susceptible to chosen ciphertext attacks.
- 16. Low iteration count in Password Based Encryption (PBE)**: Encryption key could be vulnerable to dictionary attack.
- 19. Use of key for different operations**: For security keys should have only one role.
- 33. Use of an appropriate hash function in PBKDF**: SHA-1 and MD5 are no longer considered secure.

Below the failed rules are sections for 'Passed (16 rules)' and 'Disabled (0 rules)'. The right column shows details for a selected finding:

Consequences: Compromise of the data encrypted under RSA PKCS#1v1.5

Explanation: The PKCS#1v1.5 padding mode is known to be susceptible to chosen ciphertext attacks since a paper by Bleichenbacher in 1998. In PKCS#1v2.0 (1998), it is recommended not to use PKCS#1v1.5 padding in any new applications. PKCS#1v1.5 encryption was finally removed from the TLS protocol in version 1.3.

Expertise required: High. Though the weakness is well-known, implementing the attack efficiently to suit a particular oracle demands skill.

Access required: Access to a service that responds differently when the PKCS#1v1.5 padding of a given ciphertext is valid compared to when it is invalid (a so-called padding oracle)

Resources required: Access to the oracle is the main constraint. The number of calls to the oracle to break a plaintext depends on precise details, but with the best publicly-known attack algorithm median number of calls to break a single PKCS#1v1.5 encrypted ciphertext is 15000.

Knowledge Base:

- Bleichenbacher Attack

At the bottom of the details panel, there is a 'Hide details' link. Below this is an 'Instances (1)' section showing one instance: '9.1 Wrapping with transformation: RSA/ECB/PKCS1Padding'.

The footer of the interface includes a 'FAQ' link on the left and the 'Cryptosense' logo on the right.

Vulnerability Types found by Cryptosense Analyzer

Cryptographic Usage Errors

Cryptosense Analyzer treats all the crypto operations carried out by the application under test, allowing detection of insecure combinations of operations, encryption susceptible to padding oracle attacks, reuse of one-time values, unsafe parameters, and more.

Key-Management Flaws

Thanks to our vulnerability research on common APIs Cryptosense Analyzer can precisely evaluate the security of keys protected by software keystores as well as spot other common key-management errors such as weak encryption passwords.

Algorithm and Key-Length Weaknesses

Cryptosense Analyzer detects the use of weak ciphers, hash functions, MACs and signature modes as well as short keys. Key length and algorithm policy can be customized by the user

About our Analysis Rules

Explanation of Risk

For each finding, we explain in detail the level of risk in terms of the consequences of the attack, the level of expertise required to mount it, and the computing resources required. This allows an accurate risk assessment on the basis of the threat scenario pertinent to the application under test.

Remediation Information

In addition to risk assessment information, we provide instructions on how the problem can be resolved, whether by code changes, a library update or changes to configuration files. We continually update our remediation results to take into account the complex maze of configuration files and dependencies in modern application frameworks.

Always up-to-date

Our rules are derived from academic results in applied cryptography research, standards, hacking conferences, public vulnerabilities, and our own vulnerability research. Thanks to close links to the community and ongoing collaborations with top academic groups we keep our rules up to date with latest advances in cryptanalytic attacks.

Technical Specifications

Cryptosense Analyzer supports a variety of cryptographic APIs and application environments. More are being added continually.

Cryptosense Analyzer consists of:

1. The **analysis platform** which also hosts the reporting web application, available in SaaS hosted in our cloud or as virtual machine licensed for use on-premises. The Analyzer web application works with all modern browsers, including Chrome (v55+), Firefox (v50+), Internet Explorer (11+).
2. A number of **Tracers**, which are used locally in the environment of the application under test to trace calls from the application to its cryptographic library at run-time.

Tracer Facts

- The tracer agents record the trace of calls in a file, which is compressed on the fly, and can be inspected by the user. There is no need for the agent to have a network connection to the analysis platform at the time the application under test is run. The trace can be uploaded to the analysis platform later.
- The trace contains calls to the cryptographic library including all their parameters and stack-traces to allow vulnerabilities to be pinpointed in source code.
- Traces can be obtained by leveraging existing test suites such as integration tests.
- Cryptosense supplies scripts for measuring a trace's coverage of crypto calls in the code.

Cryptosense Analyzer		
Tracer Compatibility	Cryptographic Interface	Framework Compatibility
Java		
Oracle Hotspot JVM OpenJDK, etc. version 1.6, 1.7, 1.8	Java JCA (any provider such as Oracle JCE, Bouncycastle, IBM JCE, etc.). Bouncycastle native interface is also supported.	All major application frameworks including: WebLogic Websphere Jboss/WildFly Tomcat, etc.
OpenSSL*		
Lightweight LD_PRELOAD extension, compatible OpenSSL 1.0.x and 1.1.x	Traces libssl and libcrypto (EVP interface).	
Coming soon: Microsoft .NET, python, ruby, go, and more...		

*OpenSSL currently in private beta - contact us to take part in beta programme.

For More Information
www.cryptosense.com
info@cryptosense.com

Cryptosense SA
19 Boulevard Poissonière
Paris 75002, France

France Sales: +33 (0)9 72 42 35 31
International Sales: +1 646-893-7657
sales@cryptosense.com

© 2017 Cryptosense, SA. All rights reserved. Cryptosense and the Cryptosense logo are trademarks or registered trademarks of Cryptosense SA in the U.S. and other countries. All other company and product names are the property of their respective owners.