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

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.

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.

Vulnerability Types found by Cryptosense Analyzer

Cryptographic Usage Errors
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.

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 evaluates 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 runtime.

<table>
<thead>
<tr>
<th>Crypto Interface Framework Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java JCA (any provider such as Oracle JCE, Bouncycastle, IBM JCE, etc.). Bouncycastle native interface is also supported.</td>
</tr>
<tr>
<td>All major application frameworks including: WebLogic Websphere Jboss/WildFly Tomcat, etc.</td>
</tr>
</tbody>
</table>

**Tracer Compatibility**

<table>
<thead>
<tr>
<th>Cryptographic Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Hotspot JVM OpenJDK, etc. version 1.6, 1.7, 1.8</td>
</tr>
<tr>
<td>OpenSSL* Lightweight LD_PRELOAD extension, compatible OpenSSL 1.0.x and 1.1.x</td>
</tr>
</tbody>
</table>

---

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.

---

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.

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