

---

# FIRMWARE REVIEWER V5.01 - CLOUD

## FEATURES AND FUNCTIONS OVERVIEW

This document describes the Features and Functions of Firmware Reviewer V5.01 as of May 2023.

**THE CONTENT IN THIS DOCUMENT IS INTENDED FOR USE AS PART OF A PROPOSAL DOCUMENT AND IS THEREFORE ALWAYS SUBJECT TO THE APPROPRIATE REVIEW AND APPROVAL PROCESSES TO ENSURE ACCURACY AND COMPLIANCE WITH CURRENT RESPONSE GUIDELINES. SCREENSHOTS INCLUDED AS EXAMPLES CAN CHANGE IN GRAPHIC FORMAT BUT NOT IN CONTENTS. TASK AUTOMATION CAN CHANGE DUE TO TECHNOLOGY EVOLUTIONS.**

- Overview** ..... 1
  - Firmware Reviewer – Task Automation ..... 3
  - Architecture ..... 4
- Compliance** ..... 5
- Easy to Use** ..... 6
  - Analysis Results ..... 7
  - Vendor Access ..... 8
  - Reporting ..... 9
- Firmware Detections** ..... 10
- Firmware Comparison** ..... 11
- Accuracy** ..... 12
- Firmware Reviewer Security Policy** ..... 13

# Copyright and Restricted Rights Legend

© 2015-2023 Security Reviewer Srl  
Via della Pace, 154  
Grosseto, 58015  
Italy

<https://www.securityreviewer.net>  
<https://securityreviewer.atlassian.net>

All Rights Reserved

## Notices

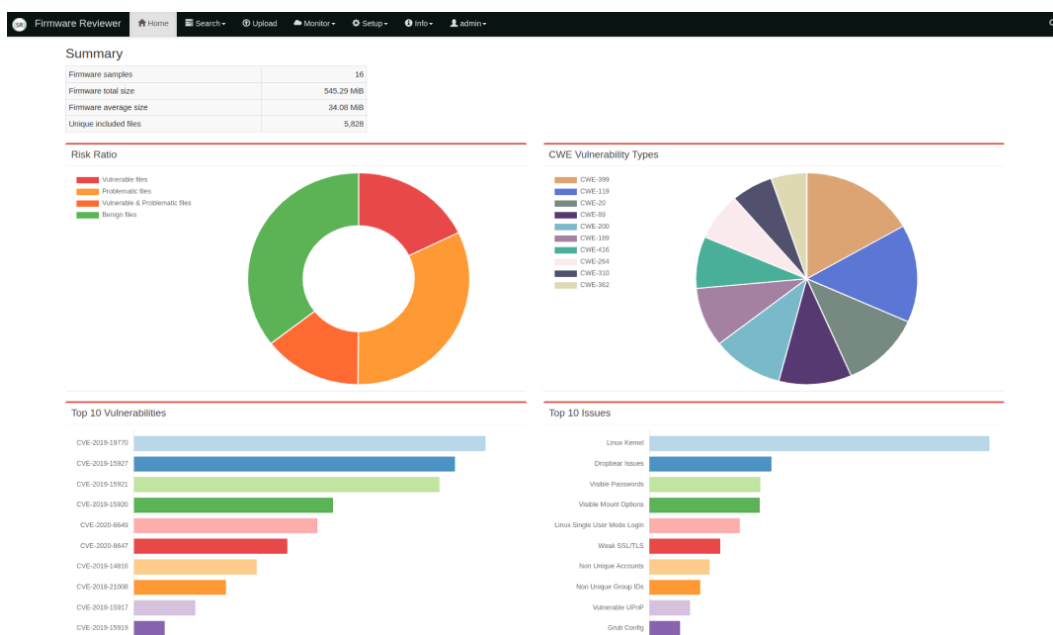
This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Security Reviewer Suite (Firmware Reviewer included) intellectual rights are registered at Italian's SIAE OLAF Office as well as at Washington Copyright Office. Other names may be trademarks of their respective owners.

# Overview

**Firmware Reviewer** Cloud Service provides in-depth firmware analysis (binaries, file systems, containers, virtual machines, IoT, Mobile, UEFI, Automotive, Network, Smart Meters, Webcams, Drones, etc.), allowing to explore vulnerabilities at the same time to keeping the software securely in your own hands, for your eyes only. It can be used for a bunch of binary file formats, with **No need of related physical device**.

Firmware Reviewer is part of [Security Reviewer Suite](#).



Firmware Reviewer does not require the Firmware source code.

Users must download the Firmware image themselves. Firmware Reviewer **never access to physical devices**.

Our Cloud infrastructure guaranteed to stay always up to date on Firmware Vulnerabilities analysis, while maintaining your data secured. See: **Firmware Reviewer Security Policy** chapter below.

Firmware Reviewer supports the following file formats: 7z, ace, apk, ar, arj, bzip2, CAB, cpio, deb, dmg, gzip, hex, ice, ipa, ISO9660, lha, lz4, lzip, LZMA, lzo,, mpkg, pkg, SFX, SREC, SY\_, rar, rpm, rzip, SIT, SQX, tar, TBZ, xar, xapk, xz, zip, zlib, zstd.from the following vendors:



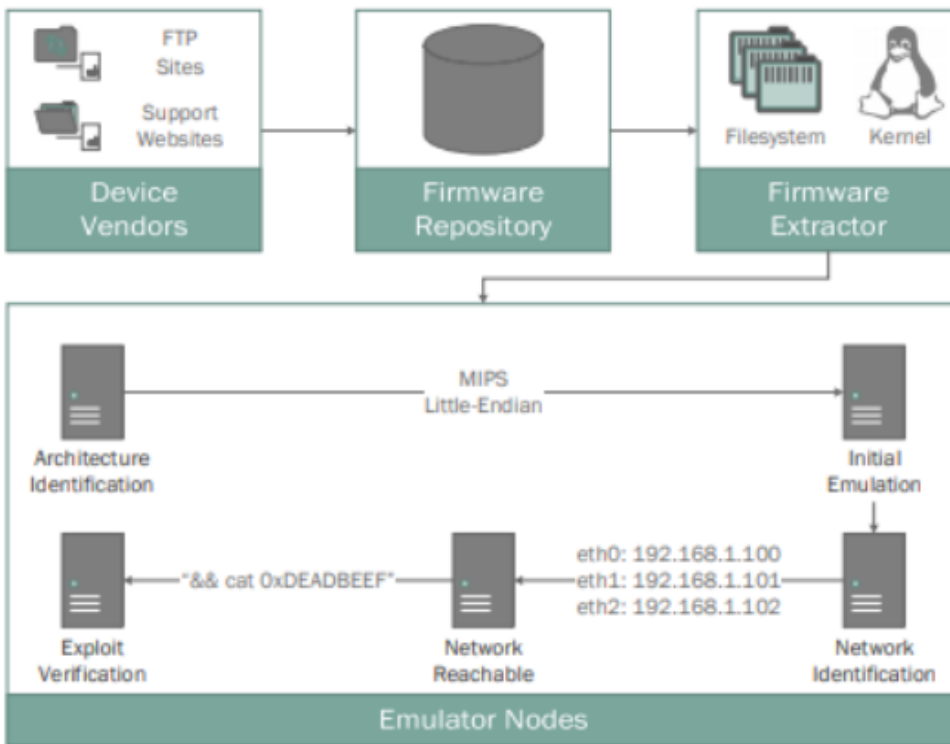
Firmware Reviewer is designed as the central firmware analysis tool for penetration testers. It supports the complete security analysis process starting with the firmware extraction process, doing static analysis and dynamic analysis via emulation, and finally generating a web report. Firmware Reviewer automatically discovers possible weak spots and vulnerabilities in firmware. Examples are insecure binaries, old and outdated software components, potentially vulnerable scripts, or hard-coded passwords. Firmware Reviewer can generate an easy-to-use web report for further analysis, that can be exported to PDF format.

Firmware Reviewer combines multiple established analysis tools and can be started with one simple click. Afterwards it tests the firmware for possible security risks and interesting areas for further investigation. No manual installation on client-side, once you have access to the Web GUI, you are ready to test your firmware.

Firmware Reviewer is designed to assist penetration testers as a standalone tool without human interaction, using **Task Automation**. Firmware Reviewer should provide as much information as possible about the firmware, that the tester can decide on focus areas and is responsible for verifying and interpreting the results.

## Firmware Reviewer – Task Automation

Firmware analysis is a tough challenge with a lot of tasks.



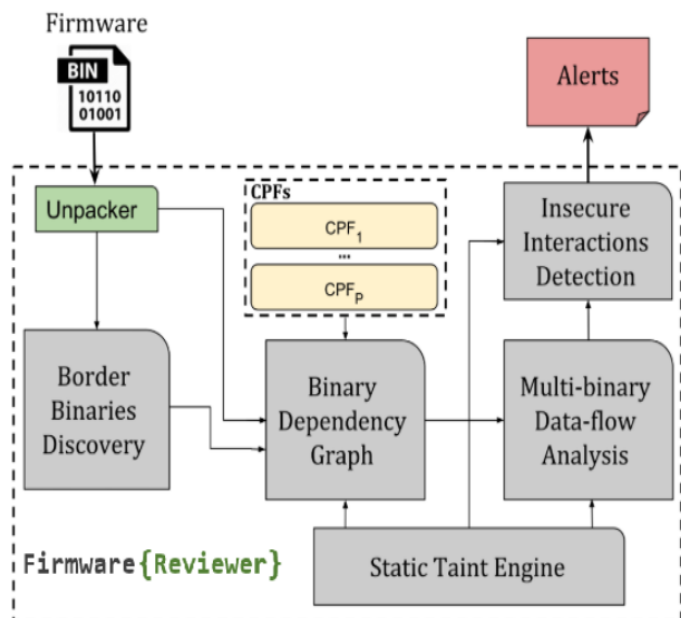
Many of these tasks can be automated (either with new approaches or incorporation of existing tools) so that a security analyst can focus on his main task: Analyzing the firmware (and finding vulnerabilities).

Firmware Reviewer					
Task					
Firmware Upload	Done	Malware Analysis	Success	Upgrade Security Alerting	Not Found
Firmware Analysis	Success	Vulnerable Components	Found	Sensitive Data	Not Found
File System Extraction	Success	Vulnerable Libraries	Found	Sensitive Data Security	Success
Credential Stored in Code	Found	Firmware Partial Emulation	Success	Anonymized Personal Data	Success
Visible Corrupt Files	Found	Firmware Complete Emulation	Not Found	Data Collecting	Success
WiFi Configurations	Found	Default Credentials	Not Found	Unencrypted Protocols	Not Found
Cipher Keys and Certificates	Found	Hardened Credentials	Not Found	Traffic Anomalies	Found
Startup Processes Analysis	Success	Password Policy	Success	Local Firewall	Success
Mounted File Systems Analysis	Success	Password Expiration	Success	Security Event Logging	Not Found
Executable Files	Found	Password Recovery	Not Found	Security Event Alerting	Found
Legacy Network Services	Found	Account Lockout	Success	Binary Fuzzing Analysis	Success
Embedded Web Server	Not Found	Role Separation	Success	Network Fuzzing Analysis	Success
Web Server Files	Not Found	Role Hardening	Success	Debug and Troubleshooting Pages	Not Found
Endpoint API	Found	Check Authentication	Success	Directory Traversal and Discovery	Found
Remote Code Execution	Found	Upgrade Feature	Not Found	Wrong Input Validation	Found
Information Disclosure	Found	Encrypted Upgrade Protocol	Not Found	Wrong Error Handling	Found
Command Injection	Found	Upgrade Validation	Not Found	Bootloader Analysis	Success
Static Analysis	Success	Upgrade Rollback	Not Found	Firmware Integrity Testing	Success

You can plan your own Tasks by choosing the ones available over 100+.

## Architecture

Firmware Reviewer is made by:



- ✚ **Front-end** Browser web GUI so that you can start right away without any further knowledge about Firmware Reviewer or the firmware you want to look at.

- ✚ **Back-end** Linux Engine. Includes an automated system for performing emulation and dynamic analysis

- ✚ **REST API** interface. Integration is easy as well since we provide a REST API covering almost all features

- ✚ **Plugin** architecture. It is based on a plug-in concept. Unpackers are implemented as plug-ins, as well as analysis features and compare functionalities

- ✚ **Alert System**

Firmware Reviewer is available in Cloud only.

frontend machine status		database machine status		backend machine status	
General		General		General	
frontend status	online	dynamic analysis status	online	dynamic analysis status	online
last status update	2020-06-06 19:17:30	last status update	2020-06-06 19:17:31	last status update	2020-06-06 19:17:29
Platform Information		Platform Information		Platform Information	
operating system	Ubuntu 16.04	operating system	Ubuntu 16.04	operating system	Ubuntu 16.04
python version	3.5.2	python version	3.5.2	python version	3.5.2
Firmware Reviewer version	3.53	Firmware Reviewer version	3.53	Firmware Reviewer version	3.53
System Stats		System Stats		System Stats	
cpu cores	4 (4 threads)	cpu cores	4 (4 threads)	cpu cores	4 (4 threads)
cpu freq	2,208.00 MHz	cpu freq	2,208.00 MHz	cpu freq	2,208.00 MHz
load average	0.25, 0.22, 0.19	load average	0.25, 0.22, 0.19	load average	0.25, 0.22, 0.19
memory usage	3.11 GiB / 11.03 GiB (31.20%)	memory usage	3.11 GiB / 11.03 GiB (31.20%)	memory usage	3.13 GiB / 11.03 GiB (31.40%)
disk usage	38.45 GiB / 217.02 GiB (18.70%)	disk usage	38.45 GiB / 217.02 GiB (18.70%)	disk usage	38.45 GiB / 217.02 GiB (18.70%)

Firmware Reviewer analyzes cyber threats on:

- Embedded Linux
- RTOS (QNX/MQX)
- VxWorks
- VMWare, QEMU, VirtualBox VMs
- Proprietary Firmware (Routers, Network HW, Radio, Mobile, Storage, Consumer IoT, etc.)

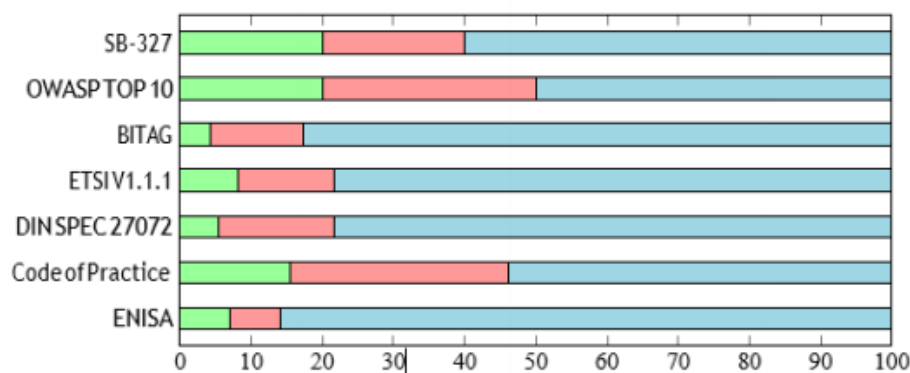
# Compliance

---

Firmware Reviewer provides reports compliant to:

- [OWASP TOP 10 INTERNET OF THINGS 2018](#)
- [ENISA - Baseline Security Recommendations for IoT](#)
- [NIST Security Feature Recommendations for IoT Devices](#)
- [DCMS GOV.UK - Code of Practice for consumer IoT security](#)
- [ETSI TS 103 645 V1.1.1 - Cyber Security for Consumer Internet of Things](#)
- [BITAG - Broadband Internet Technical Advisory Group](#)
- [SB-327 Information privacy: connected devices](#)

OWASP IoTGoat version 1.0 (ID: b75cf40730ce98d4)



**Embedded Application Security** is often not a high priority for embedded developers when they are producing devices such as routers, managed switches, medical devices, Industrial Control Systems (ICS), VoIP phones, IoT devices, and ATM Kiosks due to other challenges outside of development. Other challenges developers face may include, but are not limited to, the Original Design Manufacturer (ODM) supply chain, limited memory, a small stack, and the challenge of pushing firmware updates securely to an endpoint. Firmware Reviewer can assist you to apply [OWASP Embedded Best Practices](#), for:

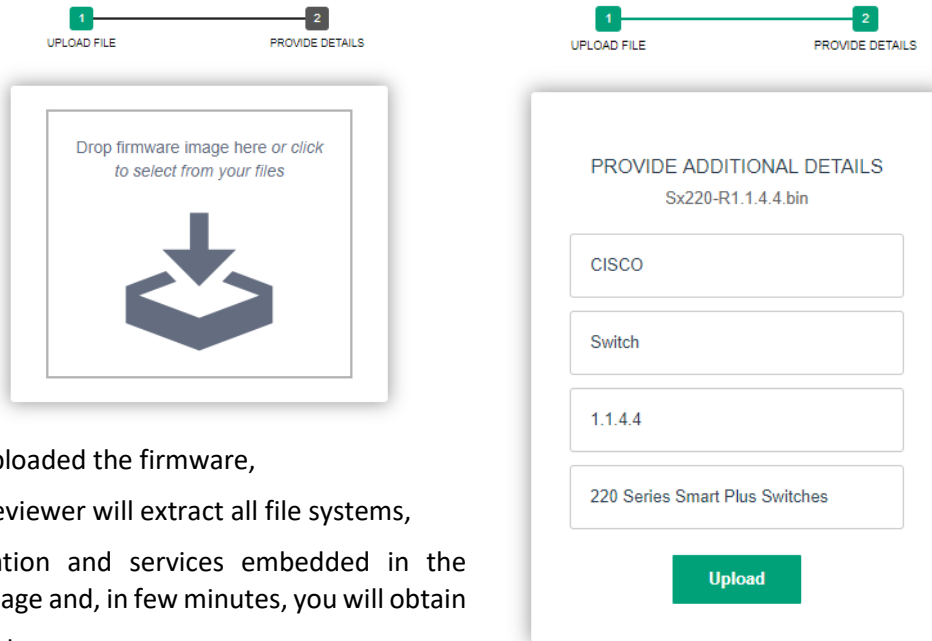
- **E1** – Buffer and Stack Overflow Protection
- **E2** – Injection Prevention
- **E3** – Firmware Updates and Cryptographic Signatures
- **E4** – Securing Sensitive Information
- **E5** – Identity Management
- **E6** – Embedded Framework and C-Based Hardening
- **E7** – Usage of Debug Code and Interfaces
- **E8** – Transport Layer Security
- **E9** – Data collection Usage and Storage – Privacy
- **E10** – Third Party Code and Components

Firmware Reviewer results are enriched with threat intelligence from [Shodan](#) and the [NIST NVD](#).

# Easy to Use

## Upload a firmware image or archive

Supported archive types: zip, tar, tar.gz



Once you uploaded the firmware, Firmware Reviewer will extract all file systems, file information and services embedded in the firmware image and, in few minutes, you will obtain the first results:

**Firmware Reviewer** Home Search Upload Monitor Setup Info admin

Comparisons +

### Analysis for OWASP IoTGoat v. 1.0

UID: b9847dd4cda69ae575bd4765423ce0882cf4b655c876cbf9a27182e3ae0fe1ab\_285736960

#### Top 10 Vulnerabilities

CVE-2019-11505	Denimsiq Issues
CVE-2019-12111	Hardcoded Passwords
CVE-2019-22109	Linux Kernel Issues
CVE-2019-12108	BusyBox Issues
CVE-2014-5461	LUA Issues
CVE-2019-8347	wpa_supplicant Issues
CVE-2019-20570	WiFi Issues
CVE-2019-18292	Vulnerable Components
CVE-2019-17261	Vulnerable Uppp
CVE-2019-19252	Non-Compliance

#### Top 10 Issues

Denimsiq Issues
Hardcoded Passwords
Linux Kernel Issues
BusyBox Issues
LUA Issues
wpa_supplicant Issues
WiFi Issues
Vulnerable Components
Vulnerable Uppp
Non-Compliance

#### Analysis Results

- Denimsiq CVE entries
- Denimsiq Multiple Buffer overflows, Denial of Service, Information Disclosure
- Hardcoded password hashes
- Linux Kernel CVE entries
- Visible conf files, Admin pages, Backdoors, IP and URI
- BusyBox CVE entries
- Lua CVE entries
- MinUPnPd CVE entries
- wpa\_supplicant CVE entries
- Compliance and legal requirements
- Features extracted
- Images visualized
- Linux ELF Analysis
- Management protocol: UPnP (Universal Plug and Play)
- Management protocol: Wi-Fi Protected Setup (WPS)
- Software component detection
- Run additional analysis

#### General

device name	IoTGoat
vendor	OWASP
device class	firmware
version	1.0
release date	2020-04-19
file name	IoTGoat-x86.img
virtual path	OWASP IoTGoat - 1.0 (firmware)
file size	272.50 MB (285,736,960 bytes)
Tags	IoT
file type	DOS boot sector

#### File Tree

```

IoTGoat-x86.img (272.50 MiB)
    
```

#### Showing Analysis: binwalk

Time of Analysis	2020-04-02 09:26:11
Plugin Version	0.5.2
Signature Analysis	

DECIMAL	HEXADESIMAL	DESCRIPTION
262144	0x40800	Linux EXT filesystem, rev 2.0, ext2 filesystem data, UID=07f8f4bc-abf4-655f-bf67-945fc9f9c0f9



# Analysis Results

You can drill-down the results:

**Analysis for OWASP IoTGoat v. 1.0**  
 UID: b9847dd4cda69ae575bd4765423ce0882cf4b655c876cbf9a27182e3ae0fe1ab\_285736960

**Top 10 Vulnerabilities**

CVE	Severity
CVE-2019-11359	Critical
CVE-2019-12111	Critical
CVE-2019-12209	Critical
CVE-2019-12308	Critical
CVE-2014-9493	Critical
CVE-2019-0747	Critical
CVE-2019-20879	Critical
CVE-2019-16295	Critical
CVE-2019-17351	Critical
CVE-2019-19252	Critical

**Top 10 Issues**

Issue	Count
Denial of Service	1
Hardcoded Passwords	1
Linux Kernel Issues	1
Busybox Issues	1
LUA Issues	1
WiFi Issues	1
Vulnerable Components	1
Vulnerable Uplink	1
Non-Compliance	1

**General**

device name	IoTGoat
vendor	OWASP
device class	firmware
version	1.0
release date	2020-04-19
file name	IoTGoat-x86.img
virtual path	OWASP IoTGoat - 1.0 (firmware)
file size	272.50 MB (285,736,960 bytes)
Tags	img
file type	DOS boot sector

**Analysis Results**

**Denial of Service CVE entries**

**13: Denial of Service**  
 Denial of Service, Denial of Service, Information Disclosure

**Hardcoded password hashes**

**Linux Kernel CVE entries**

**386: Visible conf files, Admin pages, Backdoors, IP and URI**

The firmware exposes sensitive data, such as: Configuration Files, Backdoors, Admin web pages, IP addresses and URI.

Data	Type	Description
Dropbear SSH	Service	Small SSH Client
telnet	Service	Telnet Server
sshd	Service	Small SSH Server
2.2.2	IP	Embedded IP Address
224.0.0.1	IP	Embedded IP Address
64.94.1.10.11	IP	Embedded IP Address
67.215.05.212	IP	Embedded IP Address
http://ota.wiki.kernel.org	URI	Uniform Resource Identifier
http://downloads.openwrt.org	URI	Uniform Resource Identifier
http://goesniez.net	URI	Uniform Resource Identifier
http://fab.users.org	URI	Uniform Resource Identifier
http://miniapp.free.fr	URI	Uniform Resource Identifier
Backdoor	TCP Port	Persistent Backdoor Daemon listening on port 5015
usr/lib/aa/uci/view/iotgoat/	Admin Web Pages	iotgoat.lua script shows admin web pages: function index() entry("admin", "iotgoat", "esachild", "IoTGoat", 60) dependent-false entry("admin", "iotgoat", "cwndirect", template("iotgoat/cmf"), "-", 1) entry("admin", "iotgoat", "cam", template("iotgoat/camera"), "Camera", 2) entry("admin", "iotgoat", "door", template("iotgoat/door"), "Doorlock", 3) entry("admin", "iotgoat", "webcmd", call("webcmd")) end

**Busybox CVE entries**

**Lua CVE entries**

**MiniUPnPd CVE entries**

**Analysis for OWASP IoTGoat v. 1.0**  
 UID: b9847dd4cda69ae575bd4765423ce0882cf4b655c876cbf9a27182e3ae0fe1ab\_285736960

**Top 10 Vulnerabilities**

CVE	Severity
CVE-2019-11359	Critical
CVE-2019-12111	Critical
CVE-2019-12209	Critical
CVE-2019-12308	Critical
CVE-2014-9493	Critical
CVE-2019-0747	Critical
CVE-2019-20879	Critical
CVE-2019-16295	Critical
CVE-2019-17351	Critical
CVE-2019-19252	Critical

**Top 10 Issues**

Issue	Count
Denial of Service	1
Hardcoded Passwords	1
Linux Kernel Issues	1
Busybox Issues	1
LUA Issues	1
WiFi Issues	1
Vulnerable Components	1
Vulnerable Uplink	1
Non-Compliance	1

**General**

device name	IoTGoat
vendor	OWASP
device class	firmware
version	1.0
release date	2020-04-19
file name	IoTGoat-x86.img
virtual path	OWASP IoTGoat - 1.0 (firmware)
file size	272.50 MB (285,736,960 bytes)
Tags	img
file type	DOS boot sector

**Analysis Results**

**Denial of Service CVE entries**

**13: Denial of Service**  
 Denial of Service, Denial of Service, Information Disclosure

**Hardcoded password hashes**

**Linux Kernel CVE entries**

**386: Visible conf files, Admin pages, Backdoors, IP and URI**

**Busybox CVE entries**

**Lua CVE entries**

**MiniUPnPd CVE entries**

**OWASP TOP 10 INTERNET OF THINGS 2018**

Published October 14, 2018 by OWASP - The Open Web Application Security Project Foundation. L2M (30 external refs)

**1. Insecure Default Settings**

**2. Lack of Secure Updates**

**3. Lack of Secure Boot**

**4. Lack of Secure Firmware**

**5. Lack of Secure Configuration**

**6. Lack of Secure Logging**

**7. Lack of Secure Authentication**

**8. Lack of Secure Authorization**

**9. Lack of Secure Session Management**

**10. Lack of Secure Data Storage**

**Footer: not verified**

**Images visualized**

**Linux ELF Analyse**

**Management protocol: UPnP (Universal Plug and Play)**

**Management protocol: WS-E: Protected Setup (WPS)**

**Software component detection**

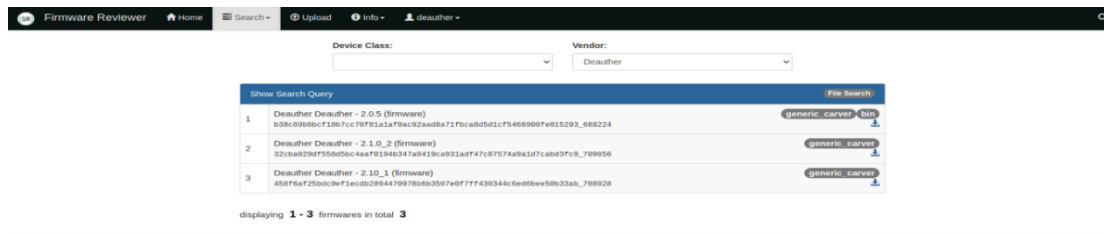
**Run additional analysis**

You can:

- In-depth browse each vulnerability
- Download all raw files, embedded in the firmware
- Download the PDF Report
- Re-do the analysis
- Update the analysis (differential)

## Vendor Access

You can give a (temporary) access to your Firmware Vendors:



**Analysis for Deauther v. 2.0.5**  
 UID: b38c89b6cf18b7cc70f81a1af0ac92aad8a71fbcab8d5d1cf5466900fe015293\_688224

**Top 10 Vulnerabilities**

CVE	Severity
CVE-2019-19770	High
CVE-2019-15927	High
CVE-2019-15921	High
CVE-2019-15920	High
CVE-2020-8649	High
CVE-2020-8647	High
CVE-2019-14816	High
CVE-2018-21008	High
CVE-2019-15917	High
CVE-2019-15919	High

**Top 10 Issues**

Issue	Severity
Linux Kernel	High
Dropbear Issues	High
Hardcoded Passwords	High
Busybox Issues	High
OpenSSL vulnerabilities	High
Authentication Bypass	High
Privilege Escalation	High
Unwanted Software	High
Non-Unique Certificates	High
Information Leakage	High

**Analysis Results**

- BusyBox CVE entries: **High**
- Dropbear SSH CVE entries: **High**
- Hardcoded password hashes: **High**
- Linux Kernel CVE entries: **High**
- OpenSSL CVE entries: **High**
- Zyrex / Huawei WIMAX CPE Authentication Bypass: **High**
- hostapd CVE entries: **Medium**
- Linux Kernel Privilege Escalation "Hail-Nelson": **Medium**
- Linux Kernel Privilege Escalation "sock\_sendpage": **Medium**
- Non-unique SSH Host Keys "House of Keys": **Medium**
- Non-unique X.509 Certificates "House of Keys": **Medium**
- Information leakage through Subversion files: **Low**
- Unwanted software: tcpdump: **Low**
- Compliance and legal requirements: **Information**
- Features extracted: **Information**
- Images visualized: **Information**
- Linux ELF Analysis: **Information**
- Management protocol: TR-069 (CPE WAN Management Protocol "CWMP"): **Information**
- Management protocol: UPnP (Universal Plug and Play): **Information**
- Management protocol: Wi-Fi Protected Setup (WPS): **Information**
- Private Keys: **Information**
- Software component detection: **Information**

Run additional analysis

The Vendors can:

- View the list of analyzed Firmware, only the related to their companies
- View every single analysis result
- Drill-down to each vulnerability
- Download raw firmware files
- Download the PDF Report


## Reporting

---

All result details will be included in an ISO-9001 compliant PDF report.

Our Reporting system is fully customizable for authorized users:

### Reports

Logo:	
ISO 9001 Template:	ISO-042020FQ1
Created by:	Davis Syman Security Reviewer
Verified by:	John Smith CAP Gemini
Approved by:	James Brown Customer Security dept.
Confidentiality Level:	<input type="radio"/> Unclassified <input type="radio"/> Internal Use Only <input type="radio"/> Confidential <input checked="" type="radio"/> Restricted <input type="radio"/> Secret
Report Password:	<input checked="" type="radio"/> None <input type="radio"/> Zip with password <input type="radio"/> Encrypted 7z
Report Obfuscation:	<input type="checkbox"/> Logo <input type="checkbox"/> Cover Letter <input type="checkbox"/> Auditor  <input type="checkbox"/> Vendor <input type="checkbox"/> File Names
<input type="button" value="Change"/>	

You can:

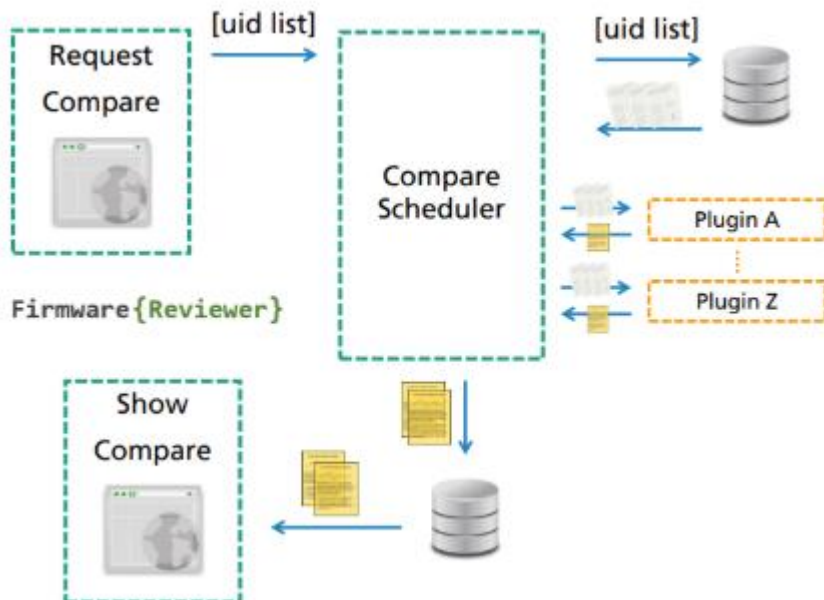
- Put your logo in all reports
- Change the report's cover letter details: ISO Template, Responsibility chain (Created, Verified, Approved)
- Set the Confidentiality Level
- Set a password to the report, or encrypt it
- Obfuscate the report, by hiding: logo, Cover letter, Auditor, Vendor or File Name, further than Credentials

## Firmware Detections

Section	Description
Device Firmware Vulnerabilities	<ul style="list-style-type: none"> <li>- Out-of-date core components</li> <li>- Unsupported core components</li> <li>- Expired and/or self-signed certificates</li> <li>- Same certificate used on multiple devices</li> <li>- Admin web interface concerns</li> <li>- Hardcoded or easy to guess credentials</li> <li>- Sensitive information disclosure</li> <li>- Sensitive IP/URI disclosure</li> <li>- Encryption key and Password hashes exposure</li> <li>- Backdoor accounts</li> <li>- Vulnerable services (web, ssh, tftp, etc.)</li> <li>- Unauthenticated access</li> <li>- Weak authentication</li> <li>- Weak Protocol (30+ supported protocols)</li> <li>- Hidden back-doors</li> <li>- Unauthenticated CGI</li> <li>- Encryption keys stored in firmware</li> <li>- Buffer overflows vulnerabilities</li> <li>- Debug services in production systems</li> </ul>
Manufacturer Recommendations	<ul style="list-style-type: none"> <li>- Ensure that supported and up-to-date software is used by developers</li> <li>- Ensure that robust update mechanisms are in place for devices</li> <li>- Ensure that certificates are not duplicated across devices</li> <li>- Ensure supported and up-to-date software is used by developers</li> <li>- Ensure a new certificate is installed when old ones expire</li> <li>- Disable deprecated SSL versions</li> <li>- Ensure developers do not code in easy to guess or common admin passwords</li> <li>- Ensure services such as SSH have a secure password created</li> <li>- Develop a mechanism that requires the user to create a secure admin password during initial device setup</li> <li>- Ensure developers do not hard code passwords or hashes</li> <li>- Have source code reviewed by a third party before releasing device to production</li> <li>- Ensure industry standard encryption or strong hashing is used</li> </ul>
Device Firmware Guidance and Instruction	<ul style="list-style-type: none"> <li>- Firmware extraction and file analysis</li> <li>- Dynamic binary analysis</li> <li>- Static binary and code analysis (40+ language supported)</li> <li>- Firmware emulation (complete, partial, sandbox)</li> <li>- File system analysis (28 supported file systems)</li> <li>- Software Composition Analysis (Third-party libraries)</li> </ul>

## Firmware Comparison

Firmware Reviewer can compare several images or single files. Furthermore, Unpacking, analysis and compares are based on plug-ins guaranteeing maximal flexibility and expandability.



In many cases you might want to compare Firmware samples. For instance, you might want to know if and where a manufacturer fixed an issue in a new firmware version. Or you might want to know if the firmware on your device is the original one provided by the manufacturer. If they differ, you want to know which parts are changed for further investigation. Again, Firmware Reviewer can automate many of these challenges, like: Identify added / changed / equal files and Identify changed software versions.

### Find other affected Firmware Images

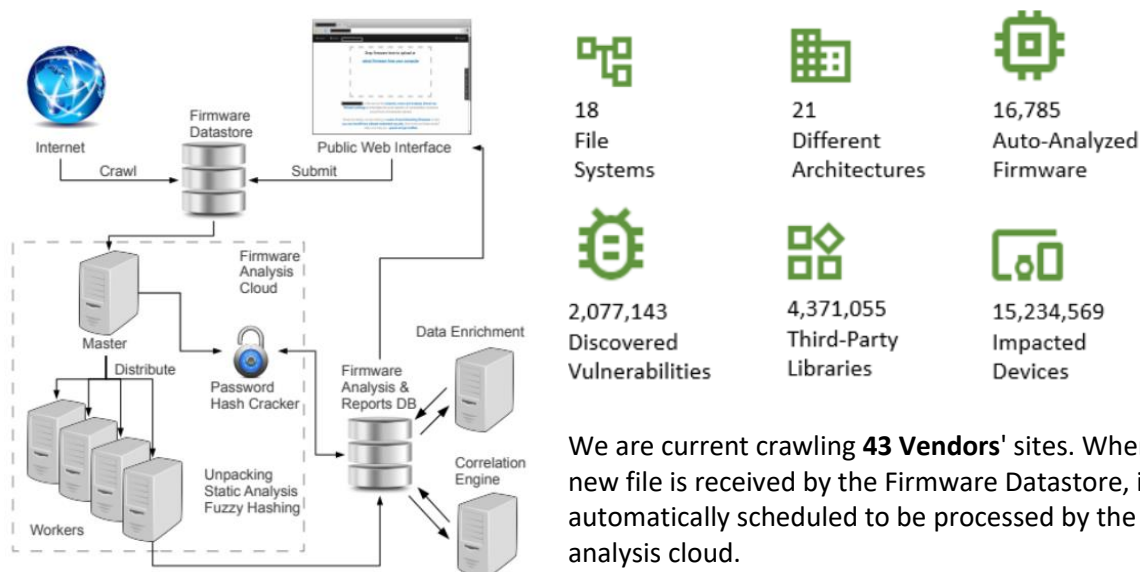
If you find a new vulnerability or a new container format, you might want to know if other firmware images share your finding. Therefore, Firmware Reviewer stores all firmware files and analysis results in a searchable database. You can search for byte patterns on all unpacked files as well as any kind of analysis result.

## Accuracy

For validating our result's accuracy, we have developed a fully automated framework and used it to test vulnerability discovery at large scale. Our system was able to find statically 38 new vulnerabilities for each of 16785 firmware packages. In addition to this, our system was able to discover dynamically 225 high-impact vulnerabilities (OWASP IoT Top Ten 2018) in at least 20% of emulated embedded web interfaces.

We also used the framework to test automated firmware and device classification. Our automated system was able to correctly classify firmware packages and identify live devices with an accuracy of 90% or more.

We explore several feature sets derived from the characteristics of firmware images, such as file size, file entropy and common strings. Then, we recommend the optimal feature set for this type of classification problems and show that our approach achieves high accuracy. Moreover, using sound statistical methods such as confidence intervals we estimate the performance of our classifiers for large scale, real world datasets. The following is an overview of the automated testing architecture:



We are current crawling **43 Vendors'** sites. When a new file is received by the Firmware Dastore, it is automatically scheduled to be processed by the analysis cloud.

In our evaluation, we used the score fusion technique to improve the accuracy of identification. The Score Fusion technique is widely and actively used in various research fields, such as biometrics and sensors data. It is used to increase the confidence in the results and to counter the effect of imprecisely approximated data (e.g., fingerprints in biometrics) and unstable data readings (e.g., sensors data). We take as input the decreasingly ordered rankings from each of the scoring systems described above. Then, we apply majority voting to each ranking from these three scoring systems. This allows our system to decide which match is the most accurate based on its scores computed using the three different scoring systems.

Our system achieved more than **90%** classification **accuracy** when the training sets were based on at least 40% of each known firmware category.

# Firmware Reviewer Security Policy

---

Firmware Reviewer Cloud Service provides in-depth firmware analysis via Web GUI. **Does not require installation on client-side.** It needs a Web Browser only.

Firmware Reviewer does not require the Firmware source code.

Users must download the Firmware image themselves. Firmware Reviewer **never access to physical devices.**

Our Cloud infrastructure guaranteed to stay always **up to date** on Firmware Vulnerabilities analysis, while maintaining your data secured.

Firmware Reviewer **does not handle Sensitive or Personal Data.** Usernames are represented by a sequence of alphanumeric characters from which is impossible to reveal information about the real Users. Once the Users got their Username and Password, they can login and Upload the Firmware Image they want to analyze.

The Firmware **Image will be encrypted** using AES-256, Uploaded using TLS 1.3 secure protocol and **stored in a crypted DB Table.**

Before Uploading, it is mandatory for the User to accept a **Disclaimer** to avoid improper use of Analysis' Results and Reports, and to confirm the User is fully authorized by the Customer and by the Vendor (the Firmware owner).

The Analysis Results will be available between 48 hours from the Upload.

Temporary files and intermediate data, generated during the analysis, even intercepted, do not permit to reverse engineering neither the Firmware Image, nor the Analysis' Results. They will be securely removed on each Analysis' step.

The Analysis' Results and Reports won't be shared to anyone else, further than authorized internal Users. They won't be visible neither fully nor partially on the Internet, neither on Social Media, nor in Electronic nor in Paper publications.

Analysis' Results and Reports will be stored in crypted DB Tables, even intercepted, it will be impossible to relate them to the original Firmware Image.

Not the Firmware Reviewer Cloud Service administrator can download Firmware Images, Results and Reports, without express, written, authorization by Customer.

Users, once the Reports has been downloaded, can decide to erase them or not. The same for Analysis' Results.