BUGPROVE

Product Security

for the Internet of Things

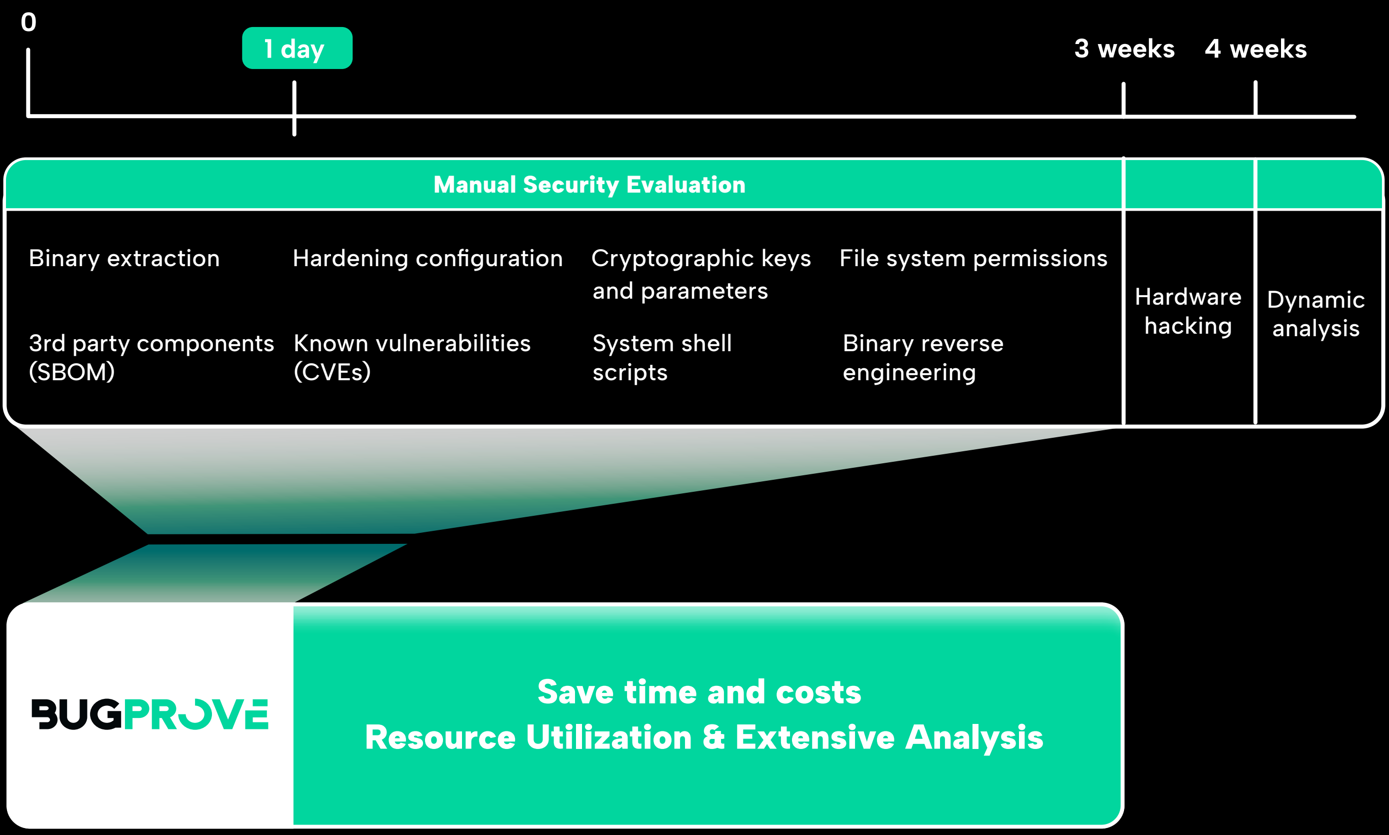
BugProve
sales@bugprove.com
https://bugprove.com/

HOW WE HELP

OUR PITCH

While the application security domain already has several available tools and solutions (like SCA, SAST, or DAST), there is still no comprehensive, out-of-the-box platform to automate security tasks for IoT devices, especially not through their entire lifecycle. Supply chain security issues are even more difficult to tackle because of the layered nature of IoT software stacks.

We offer a tool for automated zero-day and advanced CVE discovery that allows security engineers to focus on more complex problems. The tool empowers cyber-security professionals to discover common and zero-day vulnerabilities faster and more efficiently than ever before, which leads to accelerated security testing and helps you to make the most out of your value-added services.



Manual Security Evaluation					
Binary extraction	Hardening configuration	Cryptographic keys and parameters	File system permissions	Hardware hacking	Dynamic analysis
3rd party components (SBOM)	Known vulnerabilities (CVEs)	System shell scripts	Binary reverse engineering		

Save time and costs
Resource Utilization & Extensive Analysis

Manual security testing eats up engineering resources. **BugProve** saves you time to have a leaner and smoother security testing process.

BUGPROVE

HOW WE HELP

Our autonomous security testing saves you time and money as it automates the detection of critical vulnerabilities. We achieve this by pushing the boundaries of binary analysis with **integrated zero-day vulnerability discovery.** Our tool exposes security issues that are hidden in IoT products' 3rd party code as well, enabling you to **control supply chain cybersecurity risks** in a way that wasn't possible before.

Catching known vulnerabilities (CVEs) and other security issues present in SDKs is an industry need, which is the cornerstone of upcoming IoT legislations. The **SBOM** is created automatically, highlighting known vulnerabilities for outdated components.

Integrating BugProve into your testing process will support the work of security engineers daily. Our tool reduces testing slots and security administrative tasks, while our remediation recommendations and **advanced reporting** support engineers each and every project.

Our **continuous monitoring solution** automatically monitors for new threats as new vulnerabilities are disclosed. At the same time, the **upcoming compliance** verification feature collects findings and noncompliances with the industry's most sought-after standards supporting you in fulfilling your customers' specific requirements.

BOOST SECURITY ENGINEERS' EFFICIENCY WITH **BUGPROVE**'S AUTOMATED FIRMWARE ANALYSIS

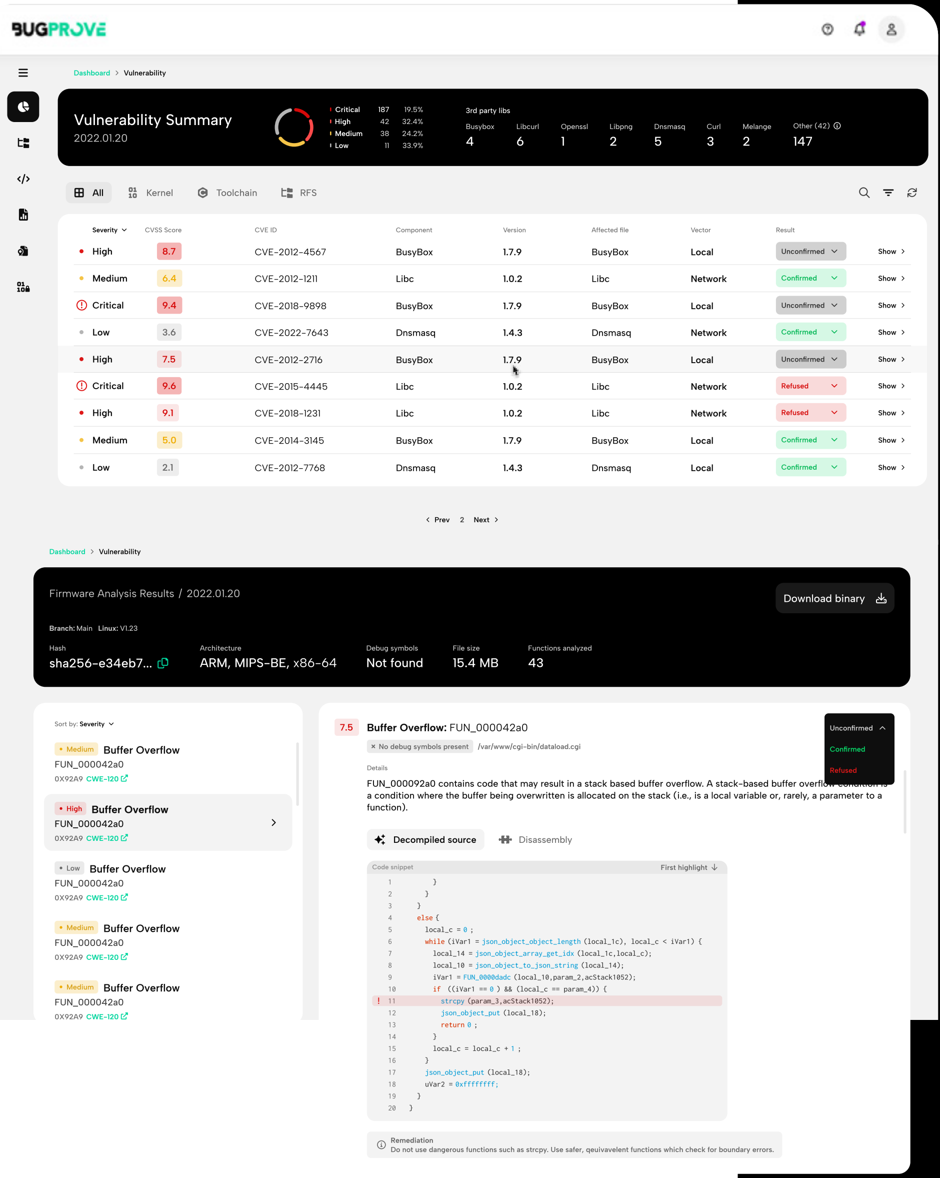
BUGPROVE

WHAT IS OUR SECRET?

UNIQUE PRODUCT CAPABILITIES

We analyze complete firmware binaries and support integrations with build pipelines based on popular tools like Buildroot and Yocto (OpenEmbedded). Our tool **can analyze entire embedded Linux firmware images, as well as individual ELF binaries,** mainstream embedded architectures like ARM, MIPS and x86-64

BugProve's scans provide an accurate, in-depth summary of **known CVEs** present in the components of your firmware. We also created a detailed dashboard to help you track these vulnerabilities at a glance and export **actionable reports** tailored to your needs. User-friendly UI makes navigating our tool a breeze, so you can stay focused on what matters the most: your product security.



Vulnerability Summary
2022.01.20

Critical	187	19.5%
High	42	32.4%
Medium	38	24.2%
Low	11	33.9%

3rd party libs

Busybox	Libcurl	Openssl	Libpng	Dnsmasq	Curl	Melange	Other (42)
4	6	1	2	5	3	2	147

Severity	CVSS Score	CVE ID	Component	Version	Affected file	Vector	Result	
High	8.7	CVE-2012-4567	BusyBox	1.7.9	BusyBox	Local	Unconfirmed	Show
Medium	6.4	CVE-2012-1211	Libc	1.0.2	Libc	Network	Confirmed	Show
Critical	9.4	CVE-2018-9898	BusyBox	1.7.9	BusyBox	Local	Unconfirmed	Show
Low	3.6	CVE-2022-7643	Dnsmasq	1.4.3	Dnsmasq	Network	Confirmed	Show
High	7.5	CVE-2012-2716	BusyBox	1.7.9	BusyBox	Local	Unconfirmed	Show
Critical	9.6	CVE-2015-4445	Libc	1.0.2	Libc	Network	Refused	Show
High	9.1	CVE-2018-1231	Libc	1.0.2	Libc	Network	Refused	Show
Medium	5.0	CVE-2014-3145	BusyBox	1.7.9	BusyBox	Local	Confirmed	Show
Low	2.1	CVE-2012-7768	Dnsmasq	1.4.3	Dnsmasq	Local	Confirmed	Show

Firmware Analysis Results / 2022.01.20

Branch: Main **Linux:** V1.23

Hash	Architecture	Debug symbols	File size	Functions analyzed
sha256-e34eb7...	ARM, MIPS-BE, x86-64	Not found	15.4 MB	43

Sort by: **Severity**

- Medium **Buffer Overflow** FUN_000042a0 0X92A9 CWE-120
- High **Buffer Overflow** FUN_000042a0 0X92A9 CWE-120
- Low **Buffer Overflow** FUN_000042a0 0X92A9 CWE-120
- Medium **Buffer Overflow** FUN_000042a0 0X92A9 CWE-120
- Medium **Buffer Overflow** FUN_000042a0 0X92A9 CWE-120

7.5 **Buffer Overflow:** FUN_000042a0

× No debug symbols present /var/www/cgi-bin/dataload.cgi

Details

FUN_000092a0 contains code that may result in a stack based buffer overflow. A stack-based buffer overflow condition is a condition where the buffer being overwritten is allocated on the stack (i.e., is a local variable or, rarely, a parameter to a function).

```
        }
      }
    }
    else {
      local_c = 0 ;
      while (iVar1 = json_object_object_length (local_1c), local_c < iVar1) {
        local_14 = json_object_array_get_idx (local_1c,local_c);
        local_10 = json_object_to_json_string (local_14);
        iVar1 = FUN_0000dadc (local_10,param_2,acStack1052);
        if  ((iVar1 == 0 ) && (local_c == param_4)) {
          strcpy (param_3,acStack1052);
          json_object_put (local_18);
          return 0 ;
        }
        local_c = local_c + 1 ;
      }
      json_object_put (local_18);
      uVar2 = 0xffffffff;
    }
  }
```

Remediation
Do not use dangerous functions such as strcpy. Use safer, qeuivavelent functions which check for boundary errors.

PROVEN TRACK RECORD

Using BugProve, vulnerabilities have already been found in the products of:

- Silicon Labs
- Honeywell
- Netgear
- Broadcom
- Asus

All our scans run on firmware images and binaries, no source code is required:

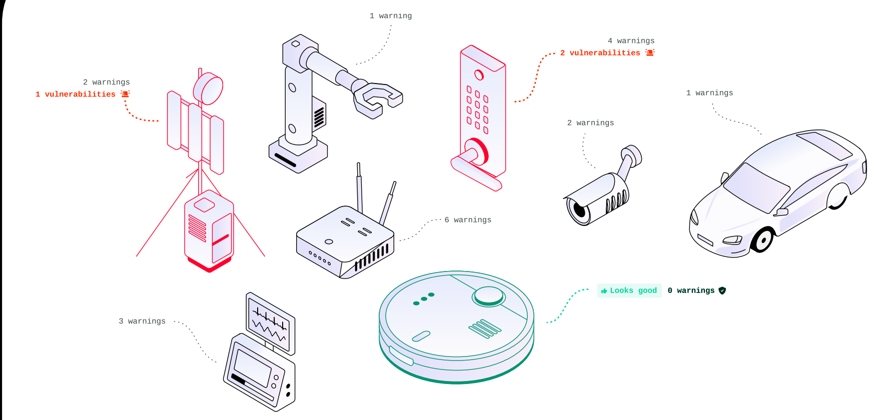
- advanced static and semi-dynamic analysis
- unique multi-binary taint analysis
- cryptographic analysis
- hardening and security configuration checks
- vulnerability **remediation**

BUGPROVE

OUR STORY

Founded in 2022, our mission is to restore stakeholder trust in embedded systems by helping IoT device manufacturers minimize security risks and improve their security posture.

The idea of BugProve dates back to the mid-2010s when the three of us worked together on embedded system security testing. It quickly became clear that the IoT sector was facing monumental challenges caused by the growing number of device vulnerabilities and malicious activities. This led us to create an automated solution to address various IoT security issues in the early stages of development, well before embedded devices are released to the market.



Eventually, Credo Ventures and Fiedler Capital – both prestigious CEE investment firms with a great track record – saw the potential in the idea. They provided the seed investment necessary to start the research and development activities in Budapest, Hungary.

We now have a dedicated team of 10+ experts with plans to expand to keep up with the growing interest surrounding our product.

LEVEL UP YOUR IOT SECURITY!

BugProve

sales@bugprove.com

https://bugprove.com/