Under the microscope:
Linux security tools

Lessons learned from 500+ projects

Michael Boelen
michael.boelen@cisofy.com
NLLGG, September 2018
Michael Boelen

- **Open Source**
  - Lynis, Rootkit Hunter

- **Business**
  - Founder of CISOfy

- **Other**
  - Blogger at Linux-Audit.com
  - Board member NLUUG
The LSE project
Project: LSE

LinuxSecurity.Expert

- Library
- People
- Toolkit
Library

- Checklists →
- Guides
- Configuration
  - sysctl
  - systemd
  - SSH
People

Profiles

● Specialists in our field
  ○ Person behind a tool
  ○ Interviews
Toolkit

- Tools
- Categories
- Snippets
Tools - Discovery
Tools - Discovery

Criteria

● Open source
● Security
● Runs on Linux, macOS, BSD
Tool analysis
# Tool analysis

<table>
<thead>
<tr>
<th>Basics</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project description</td>
<td>Changelog</td>
</tr>
<tr>
<td>Tool category</td>
<td>Popularity</td>
</tr>
<tr>
<td>Typical user</td>
<td>Documentation</td>
</tr>
<tr>
<td>License</td>
<td>Code</td>
</tr>
<tr>
<td>Author</td>
<td>Releases</td>
</tr>
<tr>
<td>Language</td>
<td>Usage</td>
</tr>
<tr>
<td>Keywords</td>
<td>Installation</td>
</tr>
<tr>
<td>Latest release</td>
<td>Ease of use</td>
</tr>
</tbody>
</table>
Tool analysis

Lynis - Security auditing tool for Linux, macOS, and UNIX-based systems. Assists with compliance testing (HIPAA/ISO27001/PCI DSS) and system hardening. Agentless, and installation optional.  [https://cisofy.com/lynis/](https://cisofy.com/lynis/)

- shell
- linux
- pci-dss
- compliance
- security-audit
- security-hardening
- security-scanner
- security-vulnerability
- hipaa
- unix
- vulnerability-detection
- vulnerability-scanners
- vulnerability-assessment
- devops
- devops-tools
- system-hardening
- hardening
- auditing
- gdpr
- security-tools
- Manage topics

- 2,158 commits
- 1 branch
- 41 releases
- 100 contributors
- GPL-3.0

Branch: master  New pull request

Superpoussin22 and mboelen detect if latest TAG is used (#575)

Latest commit 9fecefd 6 days ago
Output
Tool review

- Introduction
- Typical tool usage
- How it works
- Background details
- Strengths and weaknesses
- Example output
- Author information
- Tool alternatives
- Categories
- Tags
- And more...
## Tool and Usage

<table>
<thead>
<tr>
<th>Project details</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception</td>
<td>2007</td>
</tr>
<tr>
<td>License</td>
<td>GPLv3</td>
</tr>
<tr>
<td>Programming language</td>
<td>shell script</td>
</tr>
<tr>
<td>Author</td>
<td>Michael Boelen</td>
</tr>
<tr>
<td>Latest release</td>
<td>2.6.8 [2018-08-23]</td>
</tr>
</tbody>
</table>

### Project health

This score is calculated by different factors, like project age, last release date, etc.
# Top 100: security tools

## Tools by ranking

1. **Frida** *(reverse engineering tool)*
   - Black-box testing, reverse engineering
   - Frida allows developers and researchers to inject custom scripts into black box processes. This way it can provide a hook into any function, allowing to trace executed instructions. The source code is not needed. Frida even allows direct manipulation and see the results. The tool comes with bindings for different programming languages, allowing to interact with processes. Example of the bindings that Frida provides include Python, Swift, .NET, Qt/Qml, and C API.

   - Black box
   - Dynamic analysis
   - Reverse engineering

2. **Bro** *(network security monitoring tool)*
   - Security monitoring
   - Bro helps to perform security monitoring by looking into the network's activity. It can find suspicious data streams. Based on the data, it alert, react, and integrate with other tools.

   - IDS
   - Intrusion Detection
   - Network security monitoring
   - NIDS
   - NSM

3. **Faraday** *(collaboration tool for penetration testing)*
   - Collaboration, penetration testing, security assessment, vulnerability scanning
   - Faraday helps teams to collaborate when working on penetration tests or vulnerability management. It stores related security information in one place, which can be easily tracked and tested by other colleagues.

   - Collaboration
   - Pen testing
   - Security audit
Tools by category

Vulnerability Scanning
Lessons learned
Lessons learned - Basics

- Not really open source!
- Unclear goal
- Authorship
- Versioning
- Changelog missing
Lessons learned - Documentation

● Missing a basic description
● No ‘get started’ guide
● Lack of good examples
Lessons learned - Ease of use

- Complicated installation
- No sane defaults (e.g. --help missing)
- Parameters make no sense
What questions do you have?

Get connected

- Twitter (@mboelen and @LSELabs)
- LinkedIn (Michael Boelen)
More?

Related articles at linux-audit.com

- Why we use your open source project (or not)
- How to Promote your Open Source Project
Best Practices

--full-throttle-engine, -f
--help, -h, or help
--version, -V

Learn more: docopt.org
Best Practices

Keep a changelog

- History
- Trust
- Troubleshooting

Learn more: keepachangelog.com
Best Practices

Semantic versioning!

Major.Minor.Patch

Learn more: semver.org
Credits

Images

Where possible the origin of the used images are included in the slides. Some came without an origin from social media and therefore have no source. If you are the owner, let us know and we add the source.