

Linux Security for Developers

Insights for building a (more) secure world

Michael Boelen

michael.boelen@cisofy.com

14 January 2016

We Love Construction



Image source unknown

And Magic!

Turning data into:

- Useful output
- Stable software
- Nice services





Why Invest in Security Now?

- **Spying**
- **Internet of Things**
- **Law**
 - 2016 Dutch Data Protection Act
 - 2017-2018 European data protection law

Agenda

- What can go wrong?
- What can we do?
- Strategies and Tools

Michael Boelen

- Open Source Security
 - [Rootkit Hunter](#) (malware scan)
 - [Lynis](#) (security scan)
- 150+ blog posts at [Linux-Audit.com](#)
- Founder of [CISOfy](#)



What can go wrong?

Passwords



Image source unknown

Case: Phone House

<http://sijmen.ruwhof.net/weblog/608-personal-data-of-dutch-telecom-providers-extremely-poorly-protected-how-i-could-access-12-million-records>



Creative Users



Image source unknown

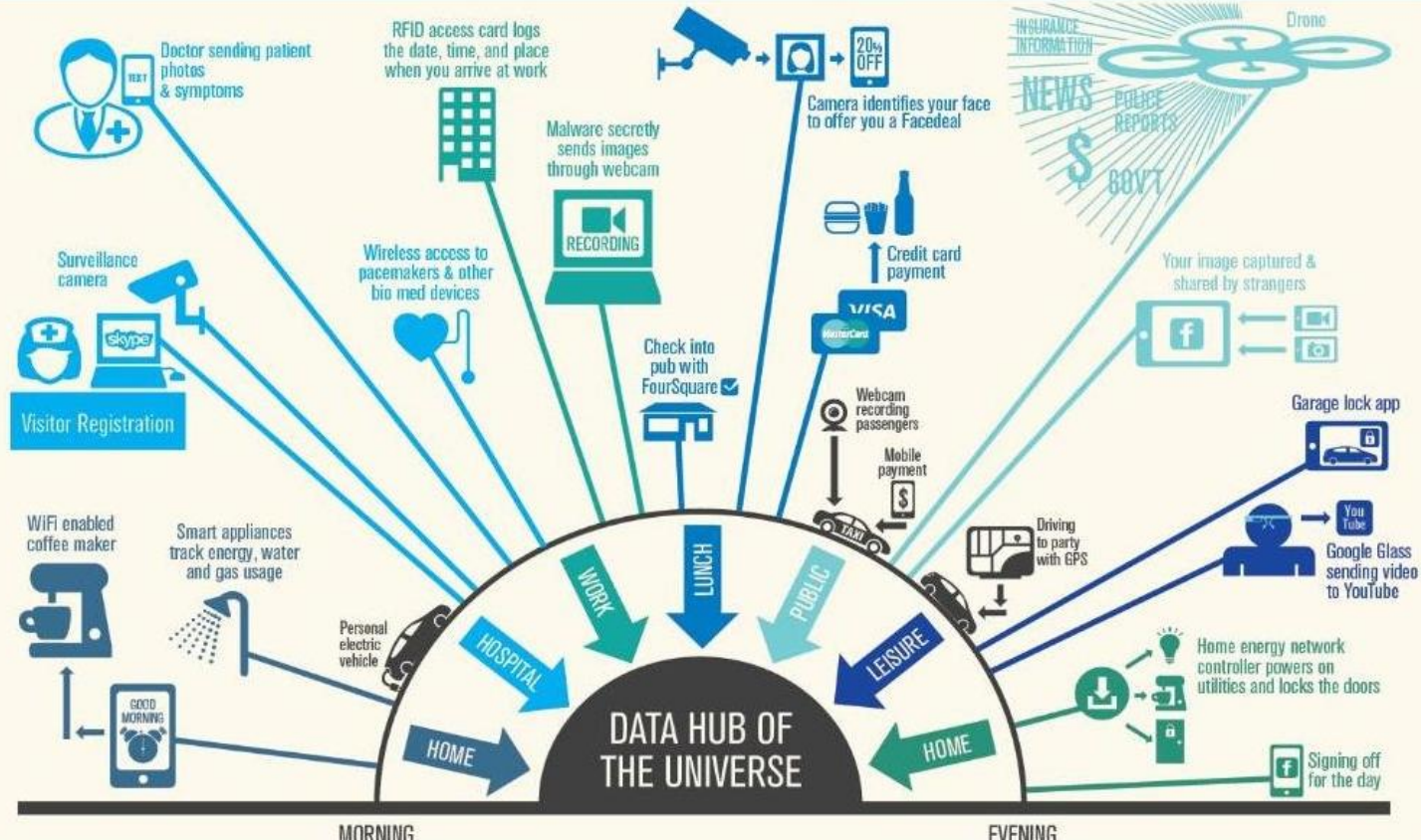
HOW MANY TIMES A DAY ARE YOU HANDING OVER YOUR INFORMATION?

From the moment we wake – and turn on that WiFi-enabled “smart” coffeemaker – to the time we make our final Facebook sign off for a long, restful sleep, we are leaving a digital trail. Most of us have no idea how the data about our daily habits, our purchases – even our routes to work – is being collected or how it’s being shared.

The infographic below outlines just a few of the hundreds of ways we voluntarily open our everyday lives to intelligence-gathering marketers, companies, government agencies, data bureaus and unknown others, simply by using our vast and growing array of technologies.

THE TAKE-AWAY?

Understand how much data you are sharing simply through every day use of gadgets and apps. Be aware of how that data may be revealing some pretty intimate details about you. If taken out of context, it may result in damaging assumptions. What can you do to lessen the data trail you leave behind every day?



What can we do?

Solution

“Developers should become auditors of their creative work, and that of others.”

Michael Boelen, 14 January 2016

What can we do?

Improve in steps

- Level 1: Basics
- Level 2: Take ownership
- Level 3: Perform auditing

Level 1: The Basics

Input Validation

Validate!

- Trust nothing
- Double check
 - **Client** = for active user
 - **Server** = for all users

Input Validation

Why Validate?

Prevent data injection (*SQL, RDF, OWL, SPARQL, SeRQL, RDQL, XML, JSON, etc.*)

Where?

Input forms, data imports

Data Protection

Encryption:

- **Good** Encryption solves a lot
- **Bad** Knowledge required
- **Ugly** Easy to implement incorrectly

Secure Programming

Using universally unique identifier?

UUID1 = Host (MAC) + sequence + time

UUID4 = Random

Two-factor Authentication

Use

- GitHub

Implement

- Your apps?

Level 2: Take Ownership

Ownership

What?

- The code
- Development systems
- Deployment
- Production

Hardening

- Add new defenses
- Improve existing defenses
- Reduce weaknesses



Photo Credits: <http://commons.wikimedia.org/wiki/User:Wilson44691>

Hardening

What to harden?

- Operating System
- Software + Configuration
- Access controls

OS Hardening

Operating System:

- Services
- Users
- Permissions

Software Hardening

Software:

- Minimal installation
- Configuration
- Tuning

Access Hardening

Users and Access Controls:

- Who can access what
- Password policies
- Accountability

Data Hardening

Focus on data streams

- Network (data in transit)
- Storage (data at rest)
- Access

Network Hardening

Traffic flows

- Is all incoming traffic needed?
- What about outgoing?
- IPv6?

HTTP Hardening

Header

X-Frame-Options SAMEORIGIN

Allow only iframe targets from our own domain

X-Frame-Options DENY

Do not allow rendering in iframe

HTTP Hardening

Header

`X-XSS-Protection 1; mode=block`

Block reflective XSS, avoid returning previous input (e.g. form)

HTTP Hardening

Header

X-Content-Type-Options nosniff

Don't peek into server responses, consider text/html by default

HTTP Hardening

securityheaders.io [Home](#) [About](#)

Scan your site now

☐ Hide results

Security Report Summary



Site: <http://linux-audit.com/> - [Scan again over https]

IP Address: 46.183.250.77

Report Time: 12 Jan 2016 20:11:21 UTC

Headers: ☒ Content-Security-Policy ☒ X-Frame-Options ☒ X-Content-Type-Options ☒ X-XSS-Protection

Raw Headers

HTTP/1.1	200 OK
Date	Tue, 12 Jan 2016 20:11:20 GMT
Content-Type	text/html; charset=UTF-8
Vary	Accept-Encoding
Link	<http://linux-audit.com/wp-json/>; rel="https://api.w.org/"
Content-Security-Policy	default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval' www.google-analytics.com; img-src 'self' cisofy.com https://*.cloudfront.net 1.gravatar.com data: ssl.google-analytics.com; style-src 'self' 'unsafe-inline' fonts.googleapis.com; font-src 'self' *.cloudfront.net data: fonts.gstatic.com; frame-src 'self'; connect-src 'self'; object-src 'none'
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
Age	0
x-cache	uncached
Transfer-Encoding	chunked

Hardening

Myth: After hardening I'm done

Server Shield v1.1.5

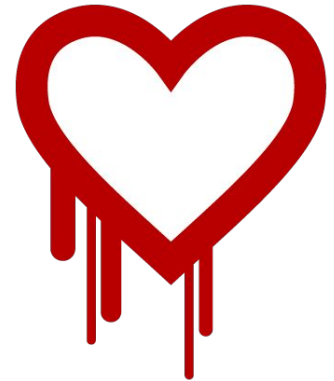
Server Shield is a lightweight method of protecting and hardening your Linux server. It is easy to install, hard to mess up, and makes your server instantly and effortlessly resistant to many basic and advanced attacks.

All IP addresses will be automatically detected and used for the firewall configuration. Automatic security updates are enabled by default.

No maintenance required— just set it and forget it!

Hardening

- Security should be an **ongoing process**
- Which means it is **never finished**
- **New attacks = more hardening**
 - [POODLE](#)
 - [Hearthbleed](#)



Level 3: Perform Auditing

Myth

Auditing =

- A lot of work!
- Booooooring!
- And.. prone to errors...

Fact

Well, it can be.

Common Strategy

1. Audit
2. Get a lot of findings
3. Start hardening
4.
- 5. Quit**

Strategy (New)

1. Focus
2. Audit
3. Focus
4. Harden
- 5. Repeat!**

1. Focus

- Determine what to scan
- Limit scope of systems / applications

2. Audit

- Start small
- Collect data

3. Focus

Determine hardening focus

- Impact
- Number
- Area (e.g. crypto)

4. Harden

- Create implementation plan
- Perform lock down
- Document
 - What, Why, How
 - Exceptions

5. Repeat

- Keep measuring your actions
- **Again:**
 - Ongoing process
 - Never finishes
 - New attacks

Questions?

Tools

Options:

1. Guides
2. Utilities



Benchmarks / Guides

- Center for Internet Security (CIS)
- NIST / NSA
- OWASP
- Vendors

Benchmarks / Guides

Pros

Free to use

Detailed

You are in control

Cons

Time intensive

Usually no tooling

Limited distributions

Delayed releases

OWASP

Open Web Application Security Project



OWASP

Security Knowledge Framework



Security Knowledge Framework

OWASP Security Knowledge Framework

The OWASP Security Knowledge Framework is intended to be a tool that is used as a guide for building and verifying secure software. It can also be used to train developers about application security. Education is the first step in the *Secure Software Development Lifecycle*.

The 4 Core usage of SKF:

- Security Requirements OWASP ASVS for development and for third party vendor applications
- Security knowledge reference (Code examples/ Knowledge Base items)
- Security is part of design with the pre-development functionality in SKF
- Security post-development functionality in SKF for verification with the OWASP ASVS


OWASP

[Link](#)

#	Description	1	2	3	Since
2.1	Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓	1.0
2.2	Verify that all password fields do not echo the user's password when it is entered.	✓	✓	✓	1.0
2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓	1.0
2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓	1.0
2.7	Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent long passphrases/highly complex passwords being entered.	✓	✓	✓	3.0
2.8	Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.	✓	✓	✓	2.0
2.9	Verify that the changing password functionality includes the old password, the new password, and a password confirmation.	✓	✓	✓	1.0
2.12	Verify that all suspicious authentication decisions are logged. This should include requests with relevant metadata needed for security investigations.		✓	✓	2.0
2.13	Verify that account passwords make use of a sufficient strength encryption routine and that it withstands brute force attack against the encryption routine.		✓	✓	3.0
2.16	Verify that credentials are transported using a suitable encrypted link and that all pages/functions that require a user to enter credentials are done so using an encrypted link.	✓	✓	✓	3.0

OWASP

← → ↻ https://www.owasp.org/index.php/OWASP_Wordpress_Security_Implementation_Guideline



Page Discussion

OWASP Wordpress Security Implementation Guideline

[hide]

- 1 Considerations
- 2 General security
 - 2.1 Device security
- 3 Infrastructure security
 - 3.1 Apache hardening
 - 3.2 PHP hardening
 - 3.3 MySQL hardening
 - 3.4 Remote access
- 4 WordPress security
 - 4.1 Updates
 - 4.1.1 WordPress Core
 - 4.1.2 Themes and Plugins
 - 4.2 Removal of unused plugins and themes
 - 4.3 Plugins & Themes Security
 - 4.4 Backup
 - 4.4.1 Automation
 - 4.5 User roles and proper usage
 - 4.6 Restrict the access to the admin interface
 - 4.7 Prevent brute-forcing
 - 4.8 Implement two factor authentication
 - 4.9 Remove or change the default administrator account
 - 4.10 Disable user registration if not needed
 - 4.11 Change the database prefix

▼ Reference

Tools

Tools

Tools make life easier, right?

Not always...

Tools

Problem 1: There aren't many

Tools

Problem 2: Usually outdated

[eglimi/linux_hardening](#)

★ 8 0

A report describing how to **harden** a **Linux** System. This work has been done as a semester project at university. It is no longer maintained and kept for reference only.

Updated on 27 Dec 2009

Tools

Problem 3: Limited support

[AdaLovelance/hardeningserverfromscratch](#)

Shell ★ 1 0

Este repositorio es un conjunto de scripts para proveer seguridad en un servidor

GNU/Linux

Updated 22 days ago

Tools

Problem 4: Hard to use

```
-<Group id="V-38581">
  <title>SRG-OS-999999</title>
  <description><GroupDescription></GroupDescription></description>
-<Rule id="SV-50382r1_rule" severity="medium" weight="10.0">
  <version>RHEL-06-000066</version>
  <title>
    The system boot loader configuration file(s) must be group-owned by root.
  </title>
  <description>
    <VulnDiscussion>The "root" group is a highly-privileged group. Furthermore, the group-owner of this file should not have any access privileges anyway.</VulnDiscussion><FalsePositives><FalsePositives><FalseNegatives><FalseNegatives>
    <Documentable>false</Documentable><Mitigations></Mitigations><SeverityOverrideGuidance><SeverityOverrideGuidance><PotentialImpacts></PotentialImpacts><ThirdPartyTools></ThirdPartyTools><MitigationControl><MitigationControl><Responsibility>
    </Responsibility><IACControls></IACControls>
  </description>
  <reference>
    <dc:title>DPMS Target Red Hat 6</dc:title>
    <dc:publisher>DISA FSO</dc:publisher>
    <dc:type>DPMS Target</dc:type>
    <dc:subject>Red Hat 6</dc:subject>
    <dc:identifier>2367</dc:identifier>
  </reference>
  <ident system="http://iase.disa.mil/cci">CCI-000366</ident>
  <fixtext fixref="F-43529r1_fix">
    The file "/etc/grub.conf" should be group-owned by the "root" group to prevent destruction or modification of the file. To properly set the group owner of "/etc/grub.conf", run the command: # chgrp root /etc/grub.conf
  </fixtext>
  <fix id="F-43529r1_fix">
  </fix>
  <check system="C-46139r1_chk">
    <check-content-ref href="DPMS_XCCDF_Benchmark_RHEL_6_STIG.xml" name="M"/>
  </check-content>
    To check the group ownership of "/etc/grub.conf", run the command: $ ls -lL /etc/grub.conf If properly configured, the output should indicate the following group-owner. "root" If it does not, this is a finding.
  </check>
  </Rule>
</Group>
```

Introducing Lynis

Lynis

Free
Open source
Shell
Simple
Flexible
Portable

```
[+] Users, Groups and Authentication
-----
- Search administrator accounts... [ OK ]
- Checking UIDs... [ OK ]
- Checking chkgrp tool... [ FOUND ]
- Consistency check /etc/group file... [ OK ]
- Test group files (grpck)... [ OK ]
- Checking login shells... [ WARNING ]
- Checking non unique group ID's... [ OK ]
- Checking non unique group names... [ OK ]
- Checking LDAP authentication support [ NOT ENABLED ]
- Check /etc/sudoers file [ NOT FOUND ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] Shells
-----
- Checking console TTYS... [ WARNING ]
- Checking shells from /etc/shells...
  Result: found 6 shells (valid shells: 6).

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] File systems
-----
- [FreeBSD] Querying UFS mount points (fstab)... [ OK ]
- Query swap partitions (fstab)... [ OK ]
- Testing swap partitions... [ OK ]
- Checking for old files in /tmp... [ WARNING ]
- Checking /tmp sticky bit... [ OK ]
```

Lynis

Background

- Since 2007
- GPLv3
- Requirements
 - Flexible
 - Portable

Lynis

Goals

- Perform a quick security scan
- Collect data
- Define next hardening steps

Lynis

Simple

- No installation needed
- Run with just one parameter
- No configuration needed

Lynis

Flexibility

- No dependencies*
- Option to extend easily
- Custom tests

* Besides common tools like awk, grep, ps

How it works

1. Initialise
2. OS detection
3. Detect binaries
4. Run helpers/plugins/tests
5. Show report

Bonus: Integration

- Deployment cycle
- Create your own tests:
include/tests_custom

Running

1. lynis
2. lynis audit system
3. lynis audit system --quick
4. lynis audit system --quick --quiet

Auditing Code

Code Validation

Quick wins

- Python: Pylint
- Ruby: ruby-lint
- Shell: shlint

Code Validation

Professional services

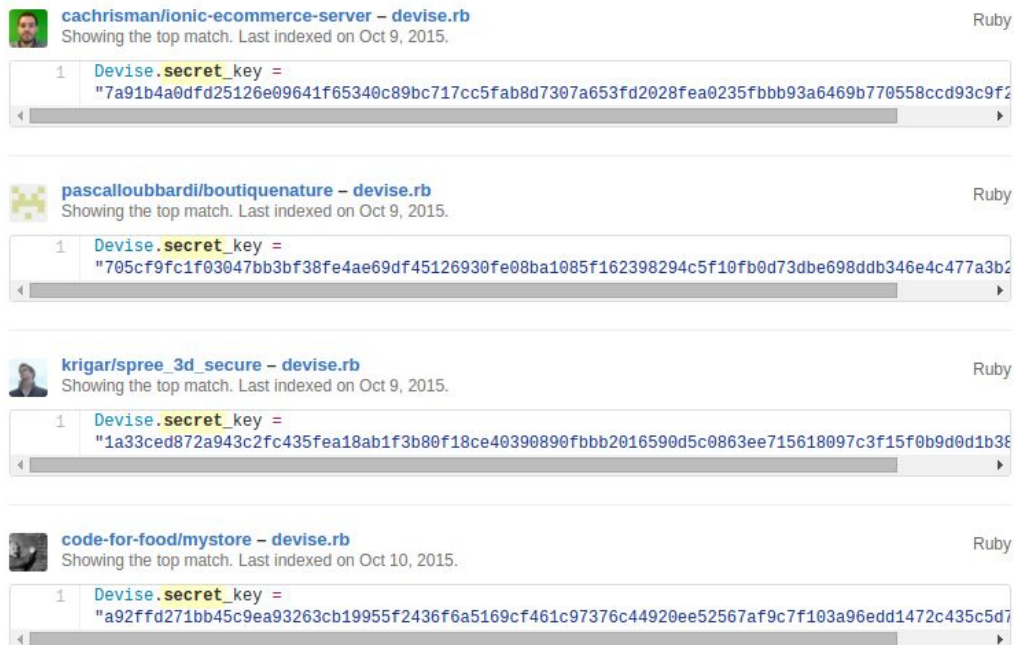
- Pentesting
- Code reviews

Auditing Repositories

Sensitive Data

- Secret keys
- Passwords
- Unique IDs
- Customers

<http://blog.arvidandersson.se/2013/06/10/credentials-in-git-repos>
<http://blog.nortal.com/mining-passwords-github-repositories/>



The screenshot displays four search results from GitHub for the query 'Devise.secret_key' in Ruby repositories. Each result shows the repository name, the file path, and the specific line of code containing the secret key. The results are as follows:

- cachrisman/ionic-ecommerce-server – devise.rb**
Showing the top match. Last indexed on Oct 9, 2015.
Line 1: `Devise.secret_key = "7a91b4a0dfd25126e09641f65340c89bc717cc5fab8d7307a653fd2028fea0235fbbb93a6469b770558ccd93c9f2"`
- pascalloubbardi/boutiquenature – devise.rb**
Showing the top match. Last indexed on Oct 9, 2015.
Line 1: `Devise.secret_key = "705cf9fc1f03047bb3bf38fe4ae69df45126930fe08ba1085f162398294c5f10fb0d73dbe698ddb346e4c477a3b2"`
- krigar/spree_3d_secure – devise.rb**
Showing the top match. Last indexed on Oct 9, 2015.
Line 1: `Devise.secret_key = "1a33ced872a943c2fc435fea18ab1f3b80f18ce40390890fbbb2016590d5c0863ee715618097c3f15f0b9d0d1b38"`
- code-for-food/mystore – devise.rb**
Showing the top match. Last indexed on Oct 10, 2015.
Line 1: `Devise.secret_key = "a92ffd271bb45c9ea93263cb19955f2436f6a5169cf461c97376c44920ee52567af9c7f103a96edd1472c435c5d7"`

Sensitive Data

Search your GitHub repos:

extension:conf password

extension:pem private

filename:..bashrc

filename:..ssh

language:ruby secret

language:python password

Hardening

Harden:

- Your systems
- Your code
- Your sensitive data

Latest Developments

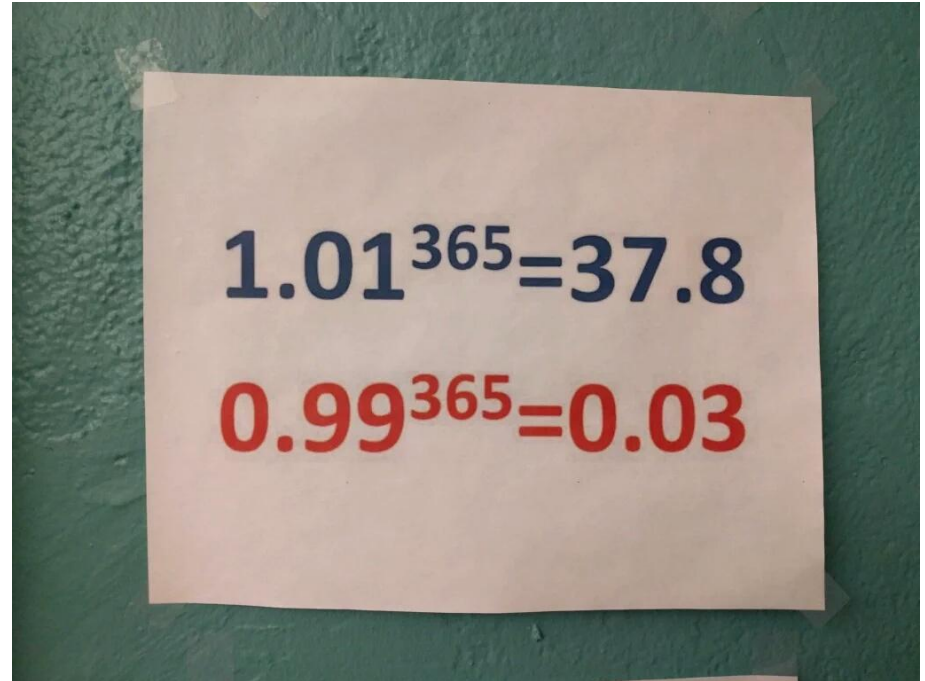
Developments

- Data protection laws
- OWASP
- New Rails security HTTP headers
- Internet of Things
- DevOps→SecDevOps / DevOpsSec

Conclusions

Lesson 1: Continuous Auditing

Many small efforts =
Big impact!



$1.01^{365} = 37.8$
 $0.99^{365} = 0.03$

Lesson 2: Implement Lynis

#include lynis.sh

```
[+] Users, Groups and Authentication
-----
- Search administrator accounts... [ OK ]
- Checking UIDs... [ OK ]
- Checking chkgrp tool... [ FOUND ]
- Consistency check /etc/group file... [ OK ]
- Test group files (grpck)... [ OK ]
- Checking login shells... [ WARNING ]
- Checking non unique group ID's... [ OK ]
- Checking non unique group names... [ OK ]
- Checking LDAP authentication support [ NOT ENABLED ]
- Check /etc/sudoers file [ NOT FOUND ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] Shells
-----
- Checking console TTys... [ WARNING ]
- Checking shells from /etc/shells...
  Result: found 6 shells (valid shells: 6).

[ Press [ENTER] to continue, or [CTRL]+C to stop ]

[+] File systems
-----
- [FreeBSD] Querying UFS mount points (fstab)... [ OK ]
- Query swap partitions (fstab)... [ OK ]
- Testing swap partitions... [ OK ]
- Checking for old files in /tmp... [ WARNING ]
- Checking /tmp sticky bit... [ OK ]
```

Lesson 3: Leverage Security

Security

- Less: Crisis and Leaks
- More: Development Time

Success!

You Finished This Presentation

Want More?

Follow Me

- Twitter: [@mboelen](https://twitter.com/mboelen)
- Personal website: michaelboelen.com
- Blog: linux-audit.com

