Don’t blink
or how to create secure software

Bozhidar Bozhanov, CEO @ LogSentinel

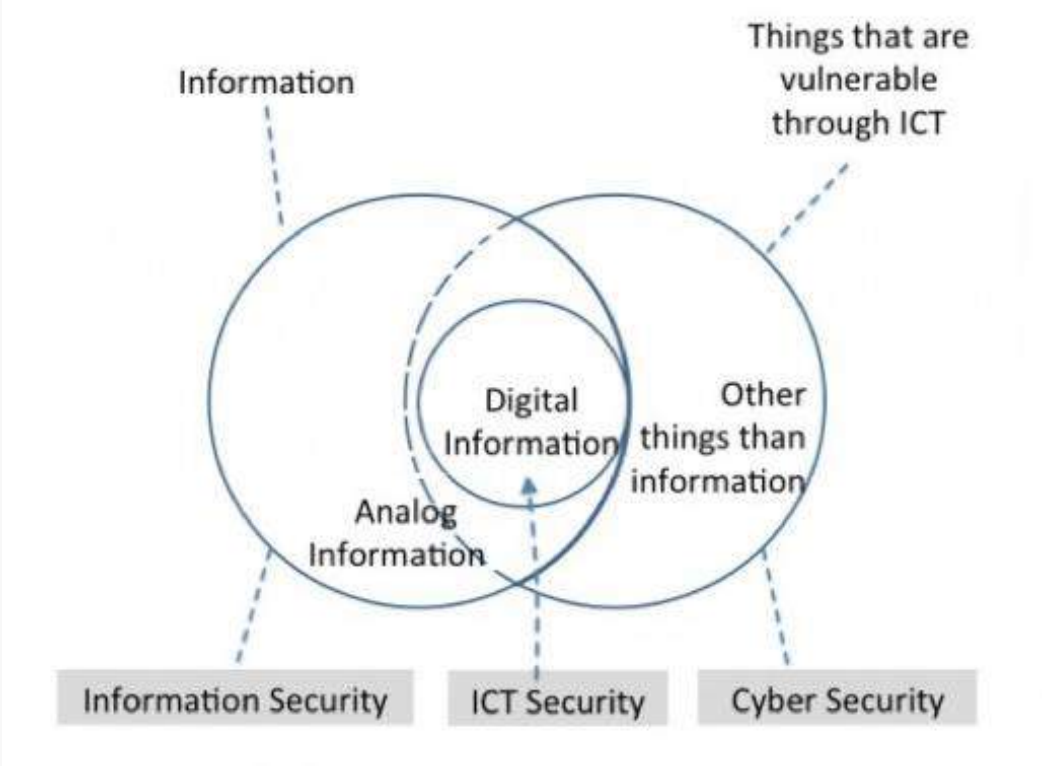
LogSentinel

About me

- Senior software engineer and architect
- Founder & CEO @ LogSentinel
- Former IT and e-gov advisor to the deputy prime minister of Bulgaria
- https://techblog.bozho.net
- Top 50 stackoverflow user
- Twitter: @bozhobg

LogSentinel

What is information security?



Source: cisoplatform.com

LogSentinel

Where can things go wrong?

LogSentinel

Purpose of attacks

Data breaches

- Breach payment information
- Breach personal data
- Breach messages (e.g. for political purposes)

Disruption

- Damaging reputation
- Stopping a time-sensitive service (e.g. promo sales, elections)
- “Some men just want to watch the world burn”

Other

- Bank transfers
- Extortion, personal motifs
- State-level espionage

Front-end vulnerabilities

XSS

- Allowing attackers to insert javascript in other people's browsers
- `<div id="comment">Unescaped input <script>alert(“hacked!")</script></div>`
- Prevention: escape all output (and filter input)

MITM

- Attacker intercepting traffic to and from server
- Can happen on any resource & lead to replacing stuff on a page
- Prevention: force HTTPS, use up-to-date cipher suites

CSRF

- Perform requests on behalf of authenticated users
- The user unknowingly submits forms
- Prevention: one-time tokens, check Origin header

Front-end vulnerability examples

British Airways

- Server with static resources compromised (modernizr.js)
- Script included in payment page
- Prevention: Limit 3rd party sources, checksums, OpSec, IPS

New South Wales e-voting

- External analytics (piwik) allowed for TLS downgrade attack
- Possible MITM to replace script and tamper votes in browser
- Prevention: Limit 3rd party sources

ES FileExplorer

- Android app, exposing the file system via HTTP
- Endangering user devices through sloppy apps
- Prevention: don't expose unauthenticated endpoints

Best practices

OWASP checklist	XSS protection	Always use HTTPS
CSRF prevention	Escape all output	Don’t store credentials on the client side
Minimize external static resources	Regularly scan static resources	Disable framing
Regularly upgrade dependencies	Get notified on known vulnerabilities	…

Back-end vulnerabilities

Remote code execution

- Allowing remote code execution
- Can allow attacker full control on entire machine/cluster
- Prevention: regularly upgrade dependencies, don't run uploads

SQL injections

- Gives full database access to attackers to extract or delete data
- `SELECT * FROM users WHERE id=1; DROP TABLE users;`
- Prevention: prepared statements

Weak authentication

- E.g. bad password storage, non-secure session cookies, etc.
- Allows attackers to impersonate users
- Prevention: review auth security, bcrypt, prevent brute force

Back-end vulnerability examples

"How I hacked 40 websites"

- Viral blogpost about a RCE attack
- Uploads a .php file as an image and invokes it via its URL
- Prevention: restrict/rename uploads, check upload contents

Bulgarian Council of Ministers

- Investigating journalists discovered SQL injection
- Old, unsupported website with personal data of civil servants
- Prevention: prepared statement / input sanitization

Marriot data breach

- Attackers got millions of passports and visitor location history
- Unencrypted sensitive data
- Prevention: encrypt data per record, IPS

Back-end best practices

Use prepared statements	Force HTTPS	Validate input on all layers
Assess your protocols	Check dependencies for vulnerabilities	Encrypt internal communication (in transit)
Encrypt sensitive data per record; bcrypt passwords	Regularly review authentication mechanisms	Don’t invent crypto stuff
Log cleverly	Keep audit trail	…

Infrastructure vulnerabilities

Credential leaks

- Credentials of privileged users leak (=insider attack)
- Allows for full control over the entire infrastructure (incl. DNS)
- Prevention: MFA, IPS, white-listed source IPs, audit trail,

Misconfiguration

- Older security suites, unauthenticated access, etc.
- Allows for direct access, downgrade attacks and others
- Prevention: penetration tests, regular reviews

Unpatched software

- Web servers, app servers, databases, OS's
- Increases the risk of 0days affecting you
- Prevention: regular upgrades, get CVE notifications

Infrastructure vulnerability examples

Heartbleed

- 0day vulnerability in OpenSSL
- All HTTPS guarantees are off, exposing any connection
- Prevention: automated patching, additional security layer

Brazilian bank DNS hijack

- DNS management credentials hacked
- All domains of the bank redirected to malicious sites
- Prevention: 2FA for all infrastructure parts, including DNS

Online check-in system

- Misconfigured web server exposing static files
- Exposed access logs containing itinerary details as GET params
- Prevention: restrict directory access, no sensitive data in GET

Infrastructure best practices

Use MFA	Configure network security rules	Monitor traffic
Keep software up-to-date	Perform penetration tests	Protect server logs
Encrypt secrets in configuration files	Use a Privileged Access Management system	Don't store production credentials in the code repos
Disable old cipher suites	Use SIEM for anomaly detection	…

LogSentinel

Human factor vulnerabilities

Spearphishing

- Target fake email
- Allows for full control over the entire infrastructure (incl. DNS)
- Prevention: MFA, IPS, email scans, audit trail

General social engineering

- Revealing personal details that can be used to extract password
- Security questions, likely passwords, etc.
- Prevention: not using personal details for authentication

Lack of training

- Non-technical employees can be targeted more easily
- Regular trainings and drills should be performed

Human factor vulnerability examples

DNC hack

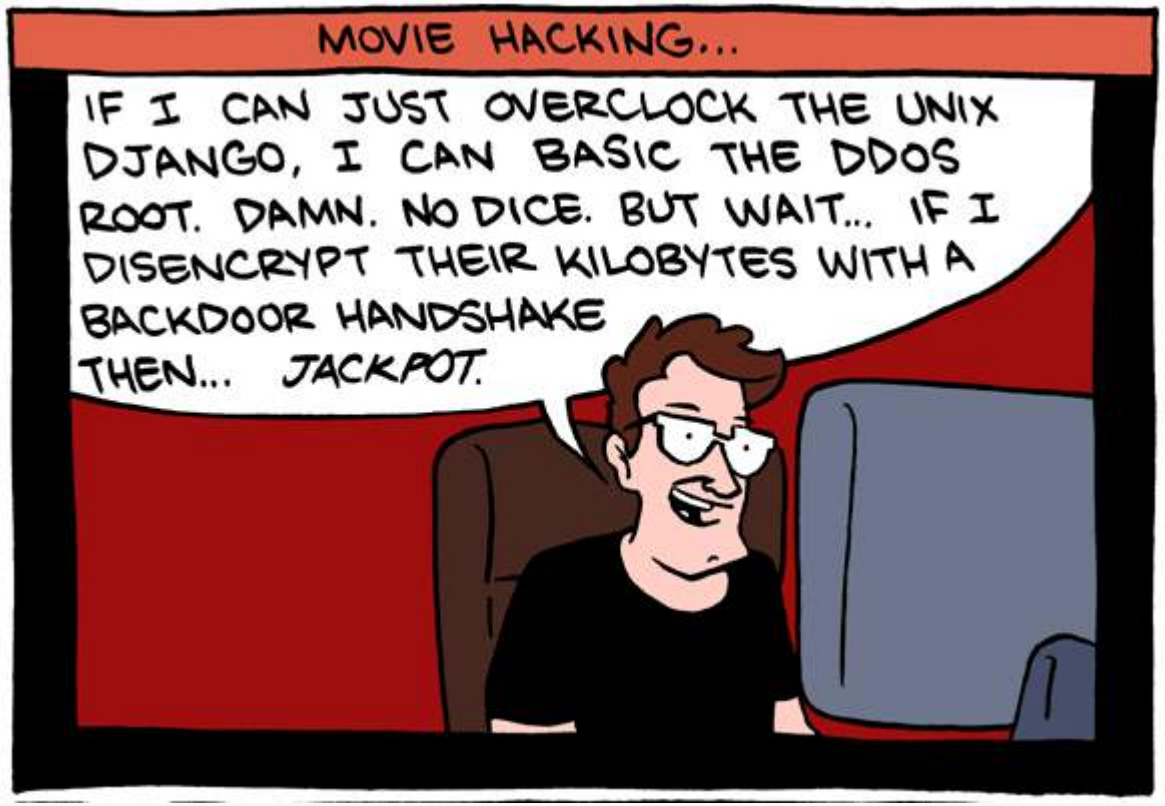
- Spearphishing email pretending to be Google
- Obtained Podesta's password to get access to the emails
- Prevention: training non-security employees, auto-detect phishing

RSA SecurID

- Spearphishing obtained secret details about SecurID 2FA fobs
- Excel attachments with Flash 0day exploit
- Prevention: training, email attachment scans

Yahoo breach

- Spearphishing a semi-privileged employee
- Attackers gained access to all Yahoo customers data
- Prevention: training, email scans, per-record data encryption



Source: http://www.smbc-comics.com/comic/2012-02-20

LogSentinel

Human factor best practices

All other best practices	Regular training of employees	Background checks (for very sensitive systems)
Audit trail	MFA	Mobile device security policies
Regular social engineering drills	Email scans	Automatic phishing detection
Limiting privileged access	Split-key access to critical systems	…

LogSentinel

So that’s it?

Obviously, nobody can give all the security knowledge in a 30 minute talk

“But we are certified!”

Good for you. ISO 27001, PCI-DSS, S-I(T)SP, etc. are a good start, but they give you the guidelines, not the implementation.

And audits happen to omit many issues

LogSentinel

A security mindset

Always ask “What can go wrong?”

Stay up-to-date

Strive for simplicity. Simple systems are more secure

Don’t assume someone else has taken care of security

A security mindset

LogSentinel

Can tools solve the problems?

Tools solve parts of the problem

Knowing your tools is hard

Tools may introduce more complexity

Look for the best tools, but not be overreliant on them

LogSentinel

Have a plan

Source: dilbert.com

LogSentinel

Bad things happen.
Be prepared to react.

Don’t do this, don’t do that,
don’t allow X, don’t enable Y,
don’t forget to do this big list of things,
don’t assume you are secure,
don’t be complacent…

So, how to create secure software?

Don’t blink!

LogSentinel

Thank you!

LogSentinel