



Security for app & services

Not a luxury but a necessity!



PAOLO ROSSI
WINTech ITALIA **CTO**

SENCHA & EMB





GITHUB PROJECTS



github.com/paolo-rossi



Delphi JWT

JSON Web Token Library



WiRL

REST Library for Delphi



Linux Daemon

Real Linux daemons



Delphi Neon

JSON Serialization Library



OpenAPI-Delphi

OpenAPI 3.0 Library



NATS Delphi

NATS Client Library for Delphi



AGENDA

1. Defining Security
2. Risks, Threats, and Vulnerabilities
3. Laws, GDPR and Application Security
4. Designing a security strategy
5. App security
6. API security



Your Security?

What is your security level?

0



WHAT IS YOUR SECURITY?

NO SECURITY

**FCS SECURITY
(FINGER CROSSED SECURITY)**



Defining Security

Section SubTitle

1



DEFINING SECURITY

- Protection of internet-connected systems
- Being protected in cyberspace means
 - ◆ Protect computer systems against unauthorized access or attack
 - ◆ Protect networks and devices from cybercrime
 - ◆ Protect data from data breaches
 - ◆ Protect applications against unauthorized access misuse



WHY IT'S IMPORTANT

Cost of a Data Breach Report 2022 (IBM)

- ◆ 83% of organizations studied have had more than 1 data breach
- ◆ 60% of breaches led to increases in prices passed on to customers
- ◆ 45% of the breaches were cloud-based
- ◆ 19% of breaches caused by stolen or compromised credentials





COSTS OF DATA BREACH

2
Middle East
2022 \$7.46

1
United States
2022 \$9.44

3
Canada
2022 \$5.64

5
Germany
2022 \$4.85

4
United Kingdom
2022 \$5.05

6
Japan
2022 \$4.57

8
Italy
2022 \$3.74

7
France
2022 \$4.34

9
South Korea
2022 \$3.57

Measured in USD millions



TYPE OF SECURITY

→ IT security

- ◆ Keeping your core information technology systems safe and intact

→ Data security

- ◆ Ensuring the integrity of all of an organization's data
- ◆ **Ensuring compliance** with data protection regulations

→ Internet of things (IoT) security

- ◆ Securing smart devices interconnected through the internet, including smartphones, laptops, tablets, etc.



TYPE OF SECURITY

- Operational technology (OT) security
 - ◆ Protecting people and assets in the monitoring of physical devices and processes
- Application security
 - ◆ Protect applications being exploited using known attacks



DEFINING DATA

- Where is the data?
- Data Inventory
 - ◆ Needs to include all data
 - ◆ Structured and unstructured,
 - ◆ Across on-premises, cloud, and third-party locations
 - ◆ Ensure that the inventory is maintained and kept up-to-date
- Data at rest vs. moving data



DATA AT REST

Type of data	Description
Public	Freely accessible to anyone (internal or external)
Internal	Intended for internal use (freely accessible)
Confidential	Sensitive information and requires authorization to access
Restricted	Critical information that could lead to severe damage to the organization should it be released



MANAGING DATA

- Probably best to separate data containers based on previous classification
 - ◆ Some of them can be fully-encrypted depending on the data value
- The separation allows an high grade of isolation and thus resistance to attacks
 - ◆ As a bonus you are better suited for an important microservice pattern



ENCRYPTING DATA

- To crypt/decrypt you use keys, in particular you need:
 - ◆ An encryption and decryption key (sometimes the same key)
 - ◆ A secure method to create an encryption key
 - ◆ A secure location to store the key
 - ◆ A way to distribute and access the key in a programmatic way
- Modern architectures can use HSM



HSM

- **Hardware Security Module**
- Hardened, tamper-resistant hardware device
- Secures cryptographic processes
 - ◆ By generating, protecting, and managing keys used for encrypting and decrypting data
- Creates digital signatures and certificates
- Usually the SDK is a DLL (SO) following the PKCS #11 API



Risks & Threats

Risks, Threats & Vulnerabilities

2



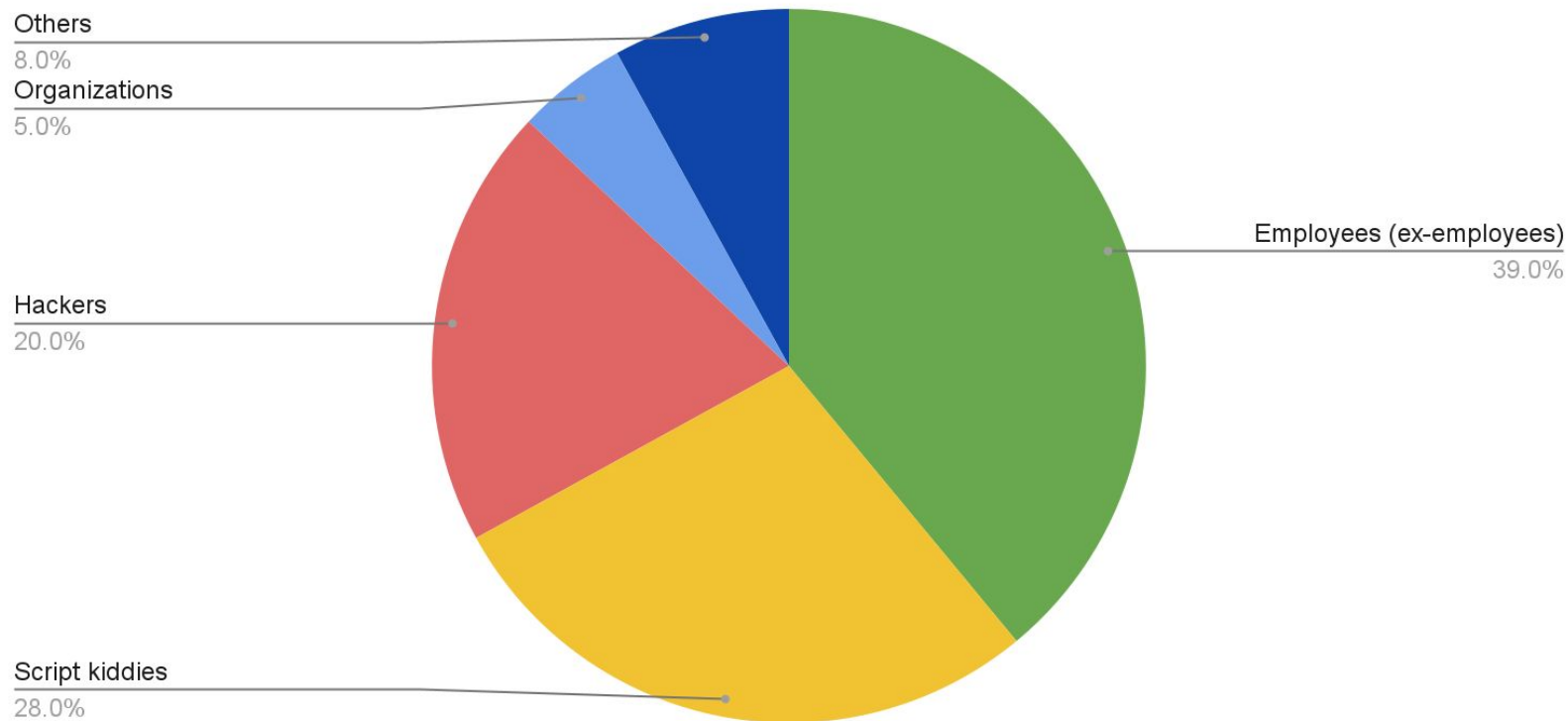
THREAT'S SOURCES

- Where the attackers come from?
- You absolutely need to know
 - ◆ In order to build defences in the right places
 - ◆ In order to choose the type of defence

CARE TO GUESS?



Threats source among organizations

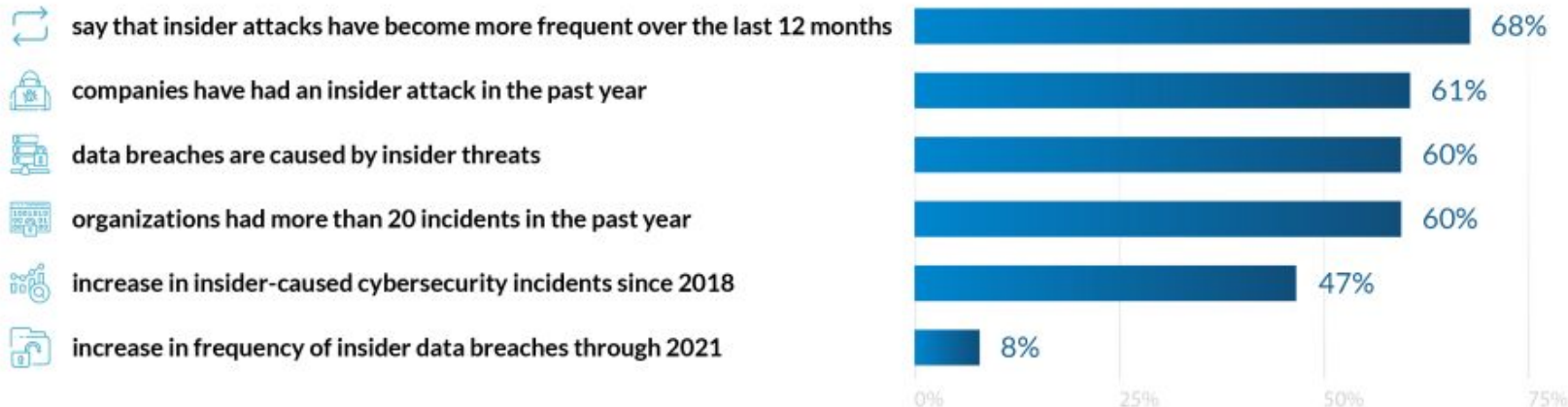




INSIDER THREAT STATISTICS

1 Insider Threat Frequency of Attacks

Sources: Goldstein, CyberSecurity, ObservelT, Shey, Bitglass, IBM





INSIDER THREAT STATISTICS

2 Top Motivations for Insider Attacks

Source: Fortinet



fraud



monetary gain



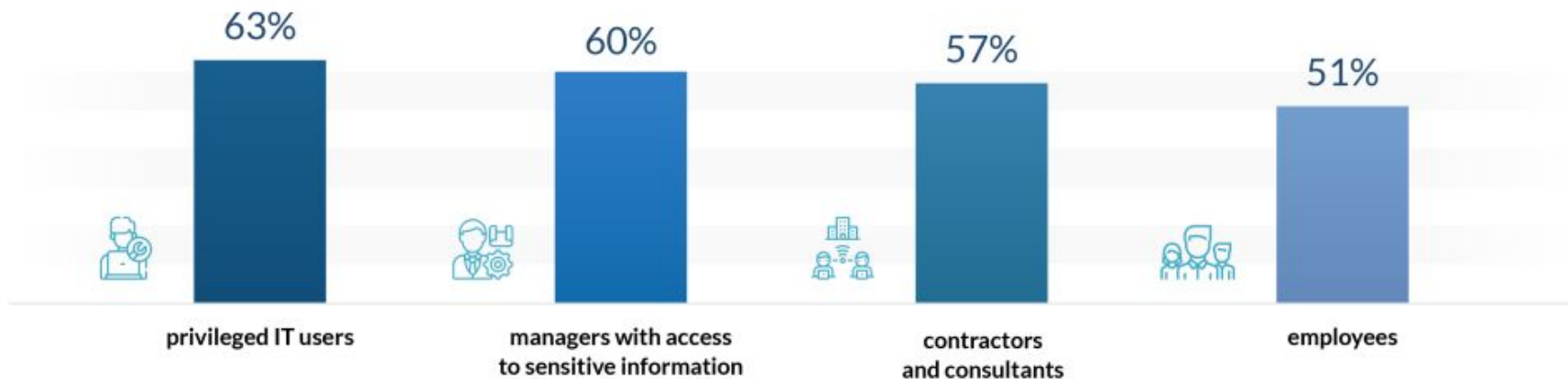
IP theft



INSIDER THREAT STATISTICS

3 Top Insider Threat Actors

Source: Cybersecurity Insers, Bitglass



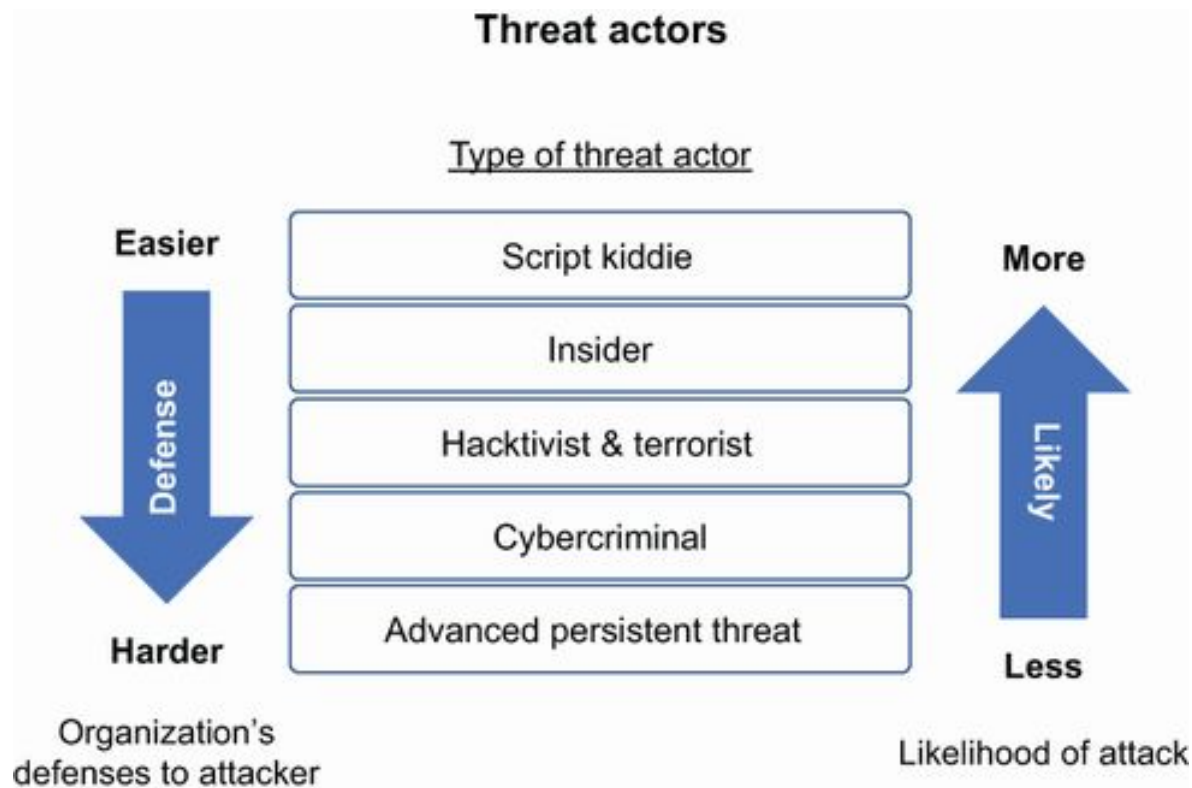


THREAT ACTORS & DEFENSE

- Knowing your adversary is key to survival
- Not all attacks have the same danger level
- Always integrate basic security (regardless of the attacker)
 - ◆ Scanning
 - ◆ Patching
 - ◆ Vulnerability management
 - ◆ Etc....



THREAT ACTORS & DEFENSE





Laws & Security

GDPR, Laws & Security

3



GDPR

- Introduced on 25 May, 2018
- Encourages organizations to properly handle user data and privacy
- Those not in compliance can face fines
- In the past years, this has resulted in more than \$1.4 billion USD in fines levied against tech giants, hotels, governmental agencies, and violators in other sectors.



GDPR REQUIREMENTS

- Enhanced application security
 - ◆ End-to-end encryption, multi-factor authentication, and so on
- Facilities for users to exercise their data privacy rights
- Quick data purge at a user's request
- Users given access to data collected from or about them
- Consent from users about what data is obtained and how it's used



SECURITY & GDPR

- GDPR can feel like a threat
- If used properly, GDPR can help you
 - ◆ It's a term that every manager knows
 - ◆ Bigger security budget
 - ◆ More security integrations
 - ◆ Faster response to security incidents



APPS & GDPR

→ Data collection

- ◆ Is there a good reason for collecting this data?
- ◆ Has the collection of non-necessary data been minimized?
- ◆ Does the user consent to the collection of this data?

→ Data breach

- ◆ Have you informed supervisors within seventy-two hours of the breach?
- ◆ Have you carried out a data protection impact assessment before adopting new technology that interacts with user data?



APPS & GDPR

→ Data protection

- ◆ Have you taken adequate measures to prevent compromise of user data?
- ◆ Are your cybersecurity policies and measures up to date?
- ◆ Is personally identifiable information (PII) of users properly encrypted to prevent abuse by malignant actors?

→ Data transfer

- ◆ Is the data transfer handled in a way that doesn't compromise data?
- ◆ Does the receiver have enough security in place to protect data?



APPS & GDPR

→ Data rights

- ◆ Do users have the ability to request copies of their data?
- ◆ Do users have the ability to object to certain uses of their data?
- ◆ Can you delete a user's data immediately if they request it?
- ◆ Does this include removing their data from your backups?



Defense Strategy

Designing a Security Strategy

4



THE NEMESIS OF SECURITY

It's always been like this



THREAT MODELING

- Identify, quantify, and address the security threats and risks associated with an application
- Fundamental parts of security
- Can be simple:
 - ◆ What could happen if a malicious user does this?
- Identify potential threats
- Identify the system or architecture
- Define the countermeasures



TERMINOLOGY

→ Attacker

- ◆ Those who intentionally or unintentionally misuse a system

→ Asset

- ◆ Anything that has value or something you must protect from an attacker

→ Threat

- ◆ A means by which an attacker might compromise an asset

→ Risk

- ◆ The potential for loss, damage, or destruction of an asset



MANUAL THREAT MODELING

What	Who	Why	How	Impact	Defense
Bank account numbers	Hackers	Financial gains	XFS	Financial payments, brand damage	X-Frame-Options, SameSite cookie attribute



SECURITY

- Security is a requirement
- Must be in the development cycle from the start
- Must be continuously checked and verified
 - ◆ Cl... anyone?
-



DETECTION TOOLS

- SAST: Static Application Security Testing
- DAST: Dynamic Application Security Testing
- IAST: Interactive Application Security Testing
- SCA: Software Composition Analysis



SAST

- Static Application Security Testing
- Code analysis on the source code looking for security issues
 - ◆ Example: Plain Text Passwords or Keys hardcoded
- Language-dependant
- Mostly as IDE plugins
- I don't know anything specific for Delphi (Pascal Analyzer?)



DAST

- Dynamic Application Security Testing
- DAST tools
 - ◆ Scan the application from the outside
 - ◆ Examine the application in its running environment
 - ◆ Manipulate the application in order to discover security vulnerabilities
- Language independent



IAST

- Interactive Application Security Testing
- Combines the strengths of SAST and DAST
- Assesses the application from within, instrumenting the code
 - ◆ Through a library to compile with the application
- Language dependant



SCA

- Software Composition Analysis
- Not all code is actually written by a developer
 - ◆ You probably use some libraries in your application
- Usually SCA tools scan open source libraries
- Language dependant



PENETRATION TESTING

- The right tool used by the right person
 - ◆ Usually performed by skilled security professional
- Internal penetration tests (performed by a red team)
- External penetration test (performed by an external party)



DEFENSE TOOLS

- RASP: Run-time Application Security Protection
- WAF: Web Application Firewall



App Security

App security countermeasures

5



APP SECURITY

- Accessi multiutente (non sai chi ha fatto cosa)
- File di log (occhio alle informazioni scritte e al livello di log)
- Database comuni dell'utente
- Password in chiaro nel database
- SQL Injection
- Password e chiavi salvate nel sorgente (e quindi nell'exe)
- File temporanei salvati sulla macchina (e sul server)
- Backdoors nel codice



API security

API security countermeasures

6



API SECURITY

- Client is irrelevant for security
 - ◆ Duplicate all validations on the server
- Testing is paramount
- How to test automatically an API?

OPENAPI 3 DOCS

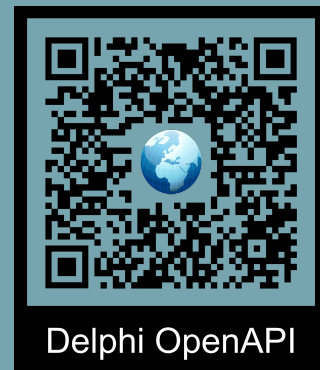


OPENAPI 3

- The OpenAPI documentation has to be auto-generated!
- If possible don't use swagger 2.0 format
- Use a REST library capable of auto-generating OpenAPI 3 docs always in sync with your code



WiRL + Delphi OpenAPI





OWASP API TOP 10 (2023)

1. Broken Object Level Authorization
2. Broken User Authentication
3. Broken Object Property Level Authorization
4. Unrestricted Resource Consumption
5. Broken Function Level Authorization
6. Server Side Request Forgery
7. Security Misconfiguration
8. Lack of Protection from Automated Threats
9. Improper Assets Management
10. Unsafe Consumption of APIs





OWASP API (2019)

- Injection
- Insufficient Logging & Monitoring





OWASP API (2019)

→ Injection

- ◆ SQL Injection, NoSQL Injection, Command Injection, etc...
- ◆ Untrusted data sent to an interpreter as part of a command or query

→ Insufficient Logging & Monitoring

- ◆ Insufficient logging and monitoring (with missing incident response)
- ◆ Attackers can further attack systems, maintain persistence, pivot to more systems to tamper with, extract, or destroy data
- ◆ Average time to detect a breach is over 200 days and typically detected by external parties (rather than internal monitoring)



#3: SERIALIZATION



→ **Broken Object Property Level Authorization**

→ Obj to JSON

- ◆ No unwanted properties
- ◆ Use ad-hoc classes (or use inheritance)

→ JSON to Obj

- ◆ Property X is not required: what if found in JSON?
- ◆ Use ad-hoc classes (or use inheritance)

→ But: always use a serializer

- ◆ Do not create JSON by hand!



#1: OBJECT LEVEL AUTH



→ **Broken Object Level Authorization**

- Exposing endpoints that handle object identifiers (ID) and not checking for the right permissions
- It's no so simple to check everything and it mudds the code
- Build an “AuthPolicy” class to be called in 1 line of code
- Complex solutions
 - ◆ Open Policy Agent (<https://www.openpolicyagent.org>)
 - ◆ OSO (<https://www.osohq.com>)



#2: USER AUTH



- **Broken User Authentication**
- Permits attackers to perform a brute force attack on the same user account, without presenting captcha/account lockout mechanism
- Permits weak passwords
- Sends sensitive authentication details, such as auth tokens and passwords in the URL



#2: USER AUTH

- Allows users to change email address, current password, without asking for password confirmation
- Doesn't validate the authenticity of token
- Accepts unsigned/weakly signed JWT tokens (`{"alg":"none"}`)
- Doesn't validate the JWT expiration date
- Uses plain text, non-encrypted, or weakly hashed passwords
- Uses weak encryption keys



#7: MISCONFIGURATION

- Latest sec. patches missing, or system out of date
- Unnecessary features enabled (HTTP verbs, logging features)
- Transport Layer Security (TLS) is missing
- Security or cache control directives are not sent to clients
- CORS policy missing or improperly set
- Error messages (to the client) include stack traces, or expose other sensitive information



THANK YOU