*This document includes disclaimers, thought process, goals, materials and core aspects of the guideline developed as a result of my research, done through a tutorial with Andrew Cencini in Computer Science, to be continued during Fall 2019.*

All examples are divided into categories. This model is aimed towards the actions a company should take to protect its users, focusing only on the business side. This means that tactics used to reduce the impact of outside elements (excluding attackers) will not be part of the technical detail, but will be mentioned and explored in terms of the responsibility and accountability of the company.

Responsibility and accountability are not referenced on a legal context - although it might be depending on the severity of the risk - but rather in an ethical approach. Regulation might or might not be part of the discussion, although the prompt for the guide was certainly regulation-oriented.

Not all points will be relevant to specific implementations. Discretion is advised when deciding which sections and points are adequate for an application, company, entreprise, or product.

Each point will have the following sections:

* What is it
* Why does it matter
* How to implement
* Pros and cons
* What can happen if I don't implement this?

The technical and narrative/explanatory information might be kept separate or presented in different formats according to the target audience. **The intention is both a call to action and the action itself,** but since these are done by different people in the ladder, the language and approach could also vary. This differentiation, if at all, will be done once the guide is complete in its original, joint format. **By informing decision makers of the dangers of their risks AND the way to solve them, awareness is raised and by extension, the likelihood of actions being taken rises.**

In points where enumeration is needed (such as security headers), this will be done for the general point and will be attempted to be done for the specific one. These sections might not be applicable for all points.

This guide is not intended to replace a security audit nor to be a complete framework. **It will serve as a supplement or general guideline for the interested parties, without warranty or guarantee.** Updates relative to new technologies and practices might or might not be done.

This guide serves two purposes. **For the administrator, it is an insight into the calculation of risks, and how to manage them.** The wording in the written sections will be such that it is easy to understand for people with a general understanding of technical terms but not of their implementation, such as managers or directors. **For the developer, architect and/or analyst, once the administrator has identified their calculated risks, it gives a proof of concept on how to implement the needed changes, if any.**

This guide has no commercial affiliation, and is done under the mantle of open access philosophy. Open source projects and free software will be cited and reference wherever possible. Commercial options will be suggested only if free software has been found insufficient for the intents and purposes.

Layers are segregated according to the general position they fall under and who is responsible for the implementation of each element. There might be intersections and shared responsibilities. A glossary of all terms in each section/subsection will follow the categories.

1. (Human) Business or Process Agents
   1. Access Control
   2. Information accessibility
   3. BYOD policies
   4. Training
      1. Technical
      2. Security
      3. Social
      4. Policy compliance
   5. Accountability and responsibility
      1. Data governance, defitinitions, roles
   6. Chain of command/hierarchy
   7. Workflow Documentation
   8. Internal policies (beyond the actual policies)
   9. Security clearance, background checks, etc
   10. Passwords
2. Physical Security
   1. Cages, server racks, appliances
   2. ID checking (clearance)
   3. Locks / Access mechanisms
   4. Cameras
   5. Shared environments (multi tenancy)
   6. IDS
3. Network
   1. Firewall
      1. Rules and policies
      2. Whitelist and blacklist
   2. WAF
      1. Embedded vs reverse proxy
   3. VPN access
   4. Port/Service restrictions (see architecture)
   5. Protocols and packets (sockets)
   6. Private vs Public access
   7. Data in Transit
   8. Honeypots
4. Architecture
   1. User authentication, access, privilege (server)
      1. Security keys, 2FA, RSA, PGP, etc
   2. Services
   3. Security Automation (see network)
   4. Activity notification
   5. Group control
   6. Directory restriction
   7. Memory management
   8. Incident Response
      1. Disclosures
      2. Mitigation
      3. Transparency
   9. Frameworks & language configuration
   10. Encapsulation & containers (multi tenancy)
       1. Server containers
       2. Kernel spaces
       3. Application (jessie frazelle - see [<https://blog.jessfraz.com/>])
   11. Failbacks & contingency
       1. Backups
   12. Sandboxing (test environments)
   13. BOM
   14. Antivirus
   15. IDS
   16. Email
       1. Encryption
       2. Association with user account
       3. Domain control
   17. Domain Records
       1. DKIM, SPF, DNS, DMARC
5. Application security
   1. Static Analysis (Compiler stuff)
   2. Secure code practices
      1. Authentication handling (keys, passwords) in files
   3. Code Review
   4. Testing (Unit, Integration)
   5. Updates, signatures and verification
   6. Input validation
   7. Standardization and canonization
   8. Access control (authorization)
   9. Authentication and passwords
      1. Database
      2. Cryptography
      3. Double blind verification
   10. Session management
       1. Cookies
       2. Sessions
       3. Tokens
   11. Layering
       1. Performance
   12. Transaction handling
   13. Error handling
   14. Logging
       1. User logs
       2. Application logs
       3. Transaction logs
   15. Privacy policies
       1. Disclosures
       2. Opt in vs opt out
       3. Third party information sharing
       4. Advertisers and business model
       5. Partnerships and sharing
   16. External Sources
   17. HTTPS and security headers (refer to SSL and TLS)
   18. API security
       1. Tokens
6. Information Storage
   1. Logs
      1. Centralized
      2. Log everything
      3. Catalog and tag logs
   2. Database
      1. Access restriction
      2. Schemas
   3. Files
      1. Uploads
      2. Integrity & checksums
   4. Version Control
   5. Backups & snapshots
      1. Expiration dates
7. Policy Compliance
   1. Choosing a framework
      1. ISO
      2. CIS (NIST)
      3. Etc
   2. Industry Regulations
   3. National (US only)
   4. International Regulations

I discussed the ideas defined above with some professionals in the Information Security field, and I got some feedback on the structure and content.