Contents

Web Security

1 Email

1.1 Architecture

The email architecture is based on several components:

- Message User Agent (**MUA**)
- Message Submission Agent (**MSA**)
- Message Transfer Agent (**MTA**)
- Message Delivery Agent (**MDA**)

In practice software implementations often merge MSA/MTA and MTA/MDA into the same component. Instead of a MUA, users tend to retrieve emails through web clients, e.g. Gmail. Emails are delivered through the **SMTP** protocol, while clients can retrieve emails in two ways:

- **POP**, where the connection is enstablished only when fetching messages, which are stored locally and then deleted from the server
- **IMAP**, where the connection is kept on while the user is browsing emails, which can be cloned locally (and optionally deleted from the server). Multiple clients can access the same mailbox

1.1.1 Message format

An email is made of three parts:

- **envelop**. Contains minimal info about sender/receiver that can be quickly processed by SMTP servers along the route. Its main fields are MAIL FROM, i.e. where to send back the message in case of failure, and RCPT TO, which is the receiver address
- **header**. Contains meta information. It is structured into fields such as FROM, TO, CONTENT-TYPE, and RETURN-PATH; may contain extra fields for security checks
- **body**. Text of the message

1.1.2 MIME

By default, SMTP supports only ASCII content. The **MIME** standard allows text in character sets other than ASCII, non-text attachments, and message bodies with multiple parts. MIME is completely transparent to the email system, since it provides a way to encode and decode this content at the endpoint.

1.2 Privacy and security risks

There are three main risks associated with emails:

- **spamming**. Unwanted or undesirable email messages
- **tracking**. The user reading an email may reveal sensitive informations, e.g. if the address is valid, IP of victim, or browser and devices used
- **phishing**. A social engineering technique where an attacker sends a "spoofed" message designed to trick a victim into revealing sensitive information. General phishing is easy to automatically detect, while spear phishing is targeted to a category of users or even a single user, making it harder to detect

1.3 Validation systems

1.3.1 SPF

One big problem with emails is that the body and the envelope of the email are somewhat separated entities. The body is built by the client that creates the email, while the envelope is made on the server through the interaction between client and server.

SPF can detect email spam through verification of the sender IP addresses. The domain publishes a DNS (TXT) record containing a list of IPs allowed to send messages, while the SMTP server validates the sender IP inside the RETURN PATH field. However it does not validate the FROM field, which can be easily spoofed, and may break when a message is forwarded. Furthermore, SPF does not validate the email content, which may be tampered.

1.3.2 DKIM

Given that SPF only works on the email envelope, **DKIM** was introduced to specify how some parts of the message can be cryptographically signed in order to avoid their content to be tampered.

Domain administrators define from which domains users are allowed to receive emails. The sender's server signs the message using his private key, while the domain published a DNS (TXT) record with info regarding his public key. Selectors can define different keys for multiple purposes/subdomains, and the receiver is then able to validate the message. However messages may be modified between server exchanges, and does not provide confidentiality.

1.3.3 DMARC

DMARC extends SPF/DKIM through policies, which define a domain email authentication practices and provide instructions on how to receive emails. In other words, domain administrator publish DNS (TXT) records with rules on what to do when a message fails DKIM/SPF and how to report abuses.

DMARC checks if the message header has proper **domain alignment**, expanding what SPF and DKIM already do. In particular, for SPF it checks if the domain in the From field correspond to the domain in the return-path field; for DKIM it checks if the domain in the From field correspond to the domain in the DKIM header field.

1.3.4 ARC

DKIM and SPF may break when a message is forwarded or altered by other SMTP servers. **ARC** provides a way to authenticate the entire chain traversed by the message, while SPF and DKIM authenticate only the original sender. This can be done using three additional headers:

- **ARC-Authentication-Results**: a combination of an instance number and the results of the SPF, DKIM, and DMARC validation
- **ARC-Seal**: A combination of an instance number, a DKIM-like signature of the previous ARC-Seal headers, and the validity of the prior ARC entries
- **ARC-Message-Signature**: A combination of an instance number and a DKIM-like signature of the entire message except for the ARC-Seal headers

Each hop in the chain signs the message.

1.4 Security for end users

1.4.1 PGP

PGP is a software which provides cryptographic privacy and authentication for data communication. It is used for signing, encrypting/decrypting texts, e-mails, and files. PGP and similar software follow the OpenPGP standard. Each public key is bound to a username or an e-mail address.

- **confidentiality**. A symmetric (session) key is generated by the sender and used to encrypt the message. This session key is encrypted with the receiver's pubblic key and is sent along with the encrypted message
- **authentication** and **integrity**. The sender computes the hash from the plaintext and then creates a digital signature using his private key. This proves that the message was not tampered and sent by the same person

There are several RFCs defining how to embed the signature and encrypted content inside a message: it can be appended as an attachment or at the end of the message. However, different clients behave in different ways. The MIME/PGP subtype mitigates this issue, but it is still kind of a mess.

Both when encrypting messages and when verifying signatures, it is critical that the public key used to send messages to someone actually does belong to the intended recipient. From its first version, PGP expands on the concept of

web of trust. Each client maintains two **keyring**: the public keyring contains all the public keys of other trusted (signed) PGP users, while the private keyring contains the public/private key pairs of the current user. Other users may also trust these keys if they trust who signed them, or if they are trusted by K users.

1.4.2 S/MIME

Extending on MIME, it provides the same security properties of PGP: authentication, message integrity, non-repudiation of origin, confidentiality. However, its trust model is based on X.509 certificates and CAs. Several RFC define new MIME subtypes, and the sender may include several certificate (other than itself) to correctly enforce the trust.

2 Web vulnerabilities

The **OWASP** initiative provides some best practices to be adapted in order to make systems more secure.

2.1 Broken Access Control

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits. Common access control vulnerabilities include:

- bypassing access control checks by modifying the URL, internal application state, or the HTML page
- permitting viewing or editing someone else's account, by providing its unique identifier
- accessing API with missing access controls for POST, PUT and DELETE
- elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user
- CORS misconfiguration allows API access from unauthorized/untrusted origins
- force browsing to authenticated pages as an unauthenticated user or to privileged pages as a standard user

Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

- except for public resources, deny by default
- minimize CORS usage

- disable web server directory listing and ensure file metadata, e.g. .git, and backup files are not present within web roots
- log access control failures, alert admins when appropriate
- rate limit API access to minimize the harm from automated attack tooling

2.2 Injection

An application is vulnerable to attack when user-supplied data is not validated, filtered, or sanitized by the application, and hostile data is directly used or concatenated. Source code review is the best method of detecting if applications are vulnerable to injections. Automated testing of parameter such as headers, URL, cookies, and input fields is strongly encouraged. Preventing injection requires keeping data separate from commands and queries:

- use server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications
- for any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter
- use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection

2.2.1 XSS

The attacker manages to inject JavaScript code, which is executed in the browser of the victim, inside a web application. As always it stems from improper sanitization of user inputs. Once users visit a comprosimed web app, they will rececive the malicious payload. A typical way to perform a XSS attack is to embed some kind of script into a snippet of html. There are three common variants of XSS attacks:

- **reflected**: the website includes data from the incoming HTTP request into the web page (without proper sanitization). Users are tricked into visiting an honest website with an URL prepared by the attacker
- **stored**: the payload is permanently stored server-side, e.g. inside the database of the web application. Whena user loads the page, his web browser executes the script
- **DOM-based**: the payload is embedded into the web page on browser-side. This means that server-side detection techniques do not work

Any user input must be preprocessed before it is used inside the page: HTML special characters must be properly encoded before being inserted into the page. Against DOM-based XSS, developers can use Trusted Types.

2.2.2 CSRF

CSRF is similar to XSS, but the attacker abuse the automatic attachment of cookies to requests done by browsers, in order to perform arbitrary actions within the session established by the victim. Assuming that the victim is authenticated on the target website:

1. the victim visits the attacker's website
2. this page triggers a request towards the victim website, e.g. forms automatically submitted via JavaScript
3. the cookie identifying this session is automatically attached by the browser

Websites can guard against CSRF by generating a token per user session, embedded inside forms (as hidden field) or as a cookie, and requiring a matching value when performing sensitive operations. This token may be generated at every page load (limiting the validity timeframe of leaked values) or once on session start (improves usability when navigating the same site).

2.3 CSP

Originally developed to mitigate content injection vulnerabilities like XSS, **CSP** can be used to restrict which resources can be loaded by a web page, and limit their origin.

2.4 SSRF

SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL). Developers can prevent SSRF by implementing some or all the following defense in depth controls:

- segment remote resource access functionality in separate networks, to reduce the impact of SSRF
- enforce "deny by default" firewall policies or network access control rules to block all but essential intranet traffic
- sanitize and validate all client-supplied input data
- enforce the URL schema, port, and destination with a positive allow list
- disable HTTP redirections

Privacy

3 Blockchain technologies

A blockchains is said to be **permitionless** if a user can join or leave the network whenever he wants, without having to be pre-approved by any central entity. There is no central owner of the network and software, and identical copies of the ledger are distributed to all the nodes in the network.

Instead a **permissioned** blockchain, transaction validators, i.e. nodes, have to be pre-selected by a network administrator (who sets the rules for the ledger) to be able to join the network. This allows to easily verify the identity of the network participants, but at the same time its participants have to put trust in the central authority to select reliable network nodes. Furthermore, there can be **open permitioned** blockchains, which can be viewed by anyone but only authorized participants can generate transactions, and **closed permitioned** blockchains, where access is restricted and only the administrator can generate transactions.

3.1 Bitcoin

The Bitcoin network is based on a **decentralized identity management**. Rather than having a central authority which registers users, anyone can generate a new **pseudo-identity** (i.e. an address) at any time. Wallets are softwares which can generate public/private key pairs, with a corresponding address. Bitcoins are sent to an address, by registering the transaction on the blockchain. Identities based on addresses can guarantee **pseudonymity**. In this context we can define anonymity as a combination of pseudonynity and **unlikability**. We can reach unlikability if the following properties hold:

- it should be hard to link together different addresses of the same user
- it should be hard to link together different transactions made by the same user
- it should be hard to link the sender of a payment to its recipient

The main idea is that we do not know who really is behind a given address. However, in practice, it's possible to deanonymize the Bitcoin network. There are services that requires a real identity, and the blockchain is public, so that any user can trace all the transactions of any other user. The "taint" of a Bitcoin transaction evaluates the association between an address and earlier transaction addresses. A possible approach to anonymize Bitcoin transactions is to use the **mixing** technique: users send their money to an anonymous service and it will send back someone else coins. However, in case of failure or if access to funds is denied, there is no recourse. An alternative solution is to have a decentralized mixing service, where different peers self-organizes in order to create the transaction, but it may not be easy to find other participants.

3.2 Zerocoin and Zerocash

Designed as an extension to the Bitcoin protocol, **Zerocoin** improves transactions anonymity by having coin-mixing capabilities natively built into the protocol. However, it is not currently compatible with Bitcoin. The underlying crypto relies on zero-knowledge proofs. In order to spend a Zerocoin, a user needs to prove that he owns that coin and it has not been already spent. Meanwhile **Zerocash** can provide additional anonymity by "shielding" the amount transacted. Unlike Zerocoin, Zerocash requires an initial setup by a trusted entity.

3.3 Other cryptocurrencies

Bitcoin cash and **Litecoin** are based on the original SHA-256 PoW algorithm with small adjustements. The foremost increased the bock size limit to $8MB$ to reduce transaction fees and improve confirmation times, while the latter offers a faster transaction speed than Bitcoin.

Instead **Algorand** achieves consensus with another kind of algorithm named Proof-of-Stakes, where a "committee" is choosen at random and its participants have to vote for a new block.

4 Privacy Preserving Data Mining

Data and knowledge extracted by data mining techniques represents a key asset to the society. Laws and regulations require that some collected data must be made public, which leads to the problem of **inference control**: protecting private data while publishing useful information

- access control: protecting information from unauthorized access and use
- disclosure control: modify data in order to prevent third-parties working with these data to recognize individuals (in the data)

In order to anonymize the data we must first remove every **Personally Identifying Information** (PII). However, an attacker may be able to join other fields with additional sources and de-anonymize the dataset. **PPDM** aims to reduce unauthorized access of private information, while retaining the same functions as a normal data mining method for discovering useful knowledge.

4.1 Microdata and macrodata

Microdata represents a series of records, containing information on an individual unit such as a person or an institution. Macrodata instead contains computed data, e.g statistics. Quasi-identifiers such as 5-digit ZIP code or gender can be used for linking anonymized dataset with other datasets, while sensitive attributes, e.g. medical records or salaries, are always released directly.

4.2 K-anonymity

Privacy risks appear when you can reidentify a record. A database is k-anonymous with respect to quasi-identifier attributes if there exist at least k records in the database having the same values. This means that individuals cannot be distinguished from $k-1$ others. This can be achieved through two techniques:

- **generalization**: replace quasi-identifiers with less specific, but semantically consistent values
- **suppression**: do not release a value at all in some sensitive tuple. Useful when generalization causes too much information loss, common with outliers

The objective is to generalize or to suppress data in a given database until it becomes k-anonymized, while incurring a minimal loss of information. The cost incurred is considerably low compared to other anonymity methods such as cryptographic solutions. However, k-anonymity this is an operational definition of a privacy mechanism rather than a mathematical definition of a privacy property; it is not much help if k individuals all possess the same sensitive attribute. It is susceptible to common attacks (unsorted matching, complementary release, homogeneity) and can cause high utility loss if it is employed in high-dimensional data and/or if the released data has already undergone anonymization more than once.

4.2.1 L-diversity and T-Completeness

L-*diversity* builds upon k-anonymity and fix one of its most obvious flaws. Let's say that all records with the same quasi-identifier tuple are in the same bucket. If all sensitive values are the same within a bucket, we might leak private information. L-*diversity* states that each bucket must have at least l distinct sensitive values. Of course, each bucket should contain at least l records, since it implies l-*anonymity*.

Even with the new added uncertainty, an attacker might have a strong suspicion about his victim sensitive information. Requiring that sensitive attributes are diverse is not enough. We need to also require that their distribution is roughly the same as the rest of the data, leading to the core idea behind t-*closeness*. This way, the attacker's knowledge can't change too much from the baseline.

However, it is hard to quantify the "amount of privacy" obtained with a given choice of parameter, and the utility loss is significant. K-*anonymity* and its variants are based on the concept of Personally Identifiable Information (**PII**), which has no guarantee and has no precise definition. These approach can still leak sensitive information, since it is assumed that attackers do not know other information about their target.

4.3 Differential privacy

In cryptography, **differential privacy** aims to provide means to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records. Since this is a probabilistic concept, any differentially private mechanism is necessarily random. For a given value of ϵ, i.e. the amount of noise introduced, there will be many differentially private algorithms for achieving the task, some with better accuracy than others.

In terms of dataset, if the effect of an arbitrary single substitution in the database is small enough, the query result cannot be used to infer much about any single individual, and therefore provides privacy. In other words, differential privacy gives each individual roughly the same privacy that would result from having their data removed.

5 Social networks

5.1 The Netflix deanonymization

Netflix, in October 2006, offered a \$1m prize for a better recommender algorithm and published the viewer ratings of $500,000$ subscribers with their names removed. Two researchers at the time showed that many subscribers could be reidentified by comparing the anonymous records with preferences publicly expressed in the IMDB. This is partly due to the "long tail" effect: once you disregard the top 100 or so movies everyone watches, people's viewing preferences are pretty unique.

5.2 Social network deanonymization

A network can be partially or totally deanonymized when the attacker, starting from a re-identication of a subset of the nodes from an auxiliary network that partially overlaps with the original network, gains knowledge about the edges of the graph (representing the network). For instance, we can consider the partial deanonymization of the Twitter graph using the Flickr graph.

5.3 Privacy concerns in social medias

People posting information by themselves on social networks increase the probability to be subject to threats such as malwares and stalking, leading to several privacy issues. All of this information about an user can be combined with social engineering techniques and lead to identity theft, be used in phishing, and other type of attacks. Often a user is not aware of the amount of information that is shared in social networks. Another problem is that creating a fake account is often easy, and can be used to lure other users in doing something wrong in order to compromise their privacy.

Furthermore users accept people as "friends" that do not know at all just to enlarge their group, but at the same time allows such people to access their

personal information and pictures. Privacy controls are within the hand of users, but the majority of users think that the information about themselves do not represent a privacy risk at all.

6 Privacy legislation

6.1 Privacy By Design

Privacy by Design (**PbD**) is an approach to system engineering which promotes privacy through the whole engineering process. This means designing data so it does not need protection. According to PbD supporters **data minimization** is the most important safeguard in protecting personally identiable information and the use of cryptography, de-identication techniques and data aggregation are absolutely critical. However PbD is similar to voluntary compliance, and its concept does not focus on the role of the actual data holder, but on that of the system designer. This role is not known in privacy law, so the concept of Privacy by Design is not based in law.

6.2 GDPR

The European Commission presented a proposal to ensure a coherent framework and a harmonized system in EU matters, which goes under the name of General Data Protection Regulations (**GDPR**). Here, two roles can be distinguished: the **data controller**, which is the person or body that, alone or jointly with others, determines the purpose and means of the processing of personal data; the **data processor**, which is a legal person or body which processes personal data on behalf of the controller.

An important aspect is that GDPR applies only for pseudonymized data. In this context, **pseudonymization** means the processing of personal data in such a way that the personal data can no longer be attributed to a specic data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organizational measure to ensure that it can not be attributed to an identifiable natural person. The GDPR defines the following characteristics of the regulations for privacy for individuals:

- **easier access to your own data**: users will have more information on how their data is processed, available in a clear and understandable way
- **right to data portability**: it will be easier to transfer your personal data between service providers
- **right to be forgotten**: when you no longer want your data to be processed, and provided that there are no legitimate grounds for retaining it, the data will be deleted

- **right to know when your data has been hacked**: companies and organizations must notify the national supervisory authority of serious data breaches as soon as possible so that users can take appropriate measures
- **data protection by design and by default**: data protection is designed into the development of business processes for products and services, and the default settings should be those that provide the most privacy
- **stronger enforcement of the rules**: data protection authorities will be able to fine companies who do not comply with EU rules up to 4% of their global annual turnover

While the characteristics of the regulations for privacy for business defined by the GDPR are the following:

- **one continent, one law**: the regulation will establish one single set of rules which will make easier and cheaper for companies to do business in the EU
- **one-stop-shop**: business will only have to deal with one single supervisory authority. It will favor the cooperation between the data protection authorities on issues of interest for all of Europe
- **european rules on european soil**: the companies outside of Europe will have to apply the same rules when offering services in EU
- **risk-based approach**: the rules will avoid a "one-size-fits-all" obligation and rather tailor them to the respective risks
- **rules fit for innovation**: data protection by design will guarantee that data protection safeguards are built into products and services from the earliest stage of development. Furthermore, privacy-friendly techniques such as pseudonymization, anonymization, and encryption will be encouraged

6.2.1 Data Protection Officer

The Data Protection Officer (**DPO**) is a designed person within an organization that collects the personal data of Union citizens, who is responsible for making sure that the organization follows the new regulations.

- inform and advise the owner of the treatment about their obligations under the European regulations
- check the implementation and application of the regulations and provide opinions on the assessment of the impact on data protection
- acts as a contact point about any issue related to the processing of their data or the exercise of their rights

A DPO must be compulsorily appointed by government departments and agencies (with the exception of judicial authorities), all person whose main business consists of treatments which require regular and systematic monitoring of those concerned, and all persons whose main business is processing sensitive data concerning health or sex life, genetic, judicial and biometrics.

6.3 Italy regulations

In Italy with the legislative decree of June 30th 2003 established the regulations and laws for protecting sensitive data according to the fast spreading of application technologies in the world. In particular, the data considered sensible by the regulations are the following:

- **sensitive data**: personal data revealing ethic origin, religious beliefs, philosophical or other beliefs, political opinions, health and sex life, etc
- **judicial data**: personal data relating to criminal records, the register of offense-related administrative sanctions and the relevant current charges, or as an accused or suspected person of the criminal procedure code

Such data needs to be processed according to the following rules:

- anyone has the right to protection of personal data concerning him
- the processing of personal data will be respect for human rights and fundamental freedoms and dignity
- the information systems and programs are congured to minimize the use of personal data and identication data

The rights of individuals data are the following:

- the right of access to personal data, including the right to obtain conrmation of the existence of data concerning him, the right to have their communication in intelligible form
- the right of cancellation, anonymization or blocking of data processed in violation of the law
- the right to oppose, for legitimate and documented reasons to the treatment

In Italy there is a collegian body composed of four members elected by Parliament called **Garante**. Such entity must communicate data processing and preliminary consent of specic actions such as the usage of detection systems of biometric, data processed with the help of electronic instruments to dene the prole or personality, etc. The major problems for companies and administrations are that they have to follow a well dened and strict set of rules regarding non disclosure of information and must have an explicit consent from the users

in order to process the data for a specic and explicit purpose. Another complications may come from **Bring your own device** approach, where companies monitors their employees together with their devices for security reasons, but this must always be done in accordance with the law.