Cloud & Application (In)security

Tejas Parikh (t.parikh@northeastern.edu)
Spring 2018

CSYE 6225
Northeastern University
https://spring2018.csye6225.com/

STAKES ARE HIGH!!!

Yahoo hackers accessed 32 million accounts with forged cookies

The company admitted execs 'failed to act' on knowledge of breaches in 2014.

Richard Lawler, @Rjcc
03.01.17 in Security

4
Comments

1053
Shares

Bloomberg via Getty Images

Verizon reportedly knocks $250 million off Yahoo's asking price

Verizon is close to renegotiating its deal to acquire Yahoo's internet assets in light of recent hacks and related controversies, Bloomberg said.

Tech Industry

by **Richard Nieva**
February 15, 2017 7:39 AM PST
@richardjnieva

by **Ry Crist**
February 15, 2017 7:39 AM PST
@rycrist

Apache Struts Vulnerability Takes Down Canadian Tax Agency

By Sean Michael Kerner | Posted 2017-03-14 Print

Tweet LinkedIn 8 Like 2 G+ Share 2 Share 8 Email

The Government of Canada reveals that one of its agency sites was breached by attackers making use of the recently disclosed Apache Struts vulnerability.

The open-source Apache Struts project first disclosed a high impact critical remote code execution vulnerability on March 6 and now it has claimed its first public victim. The Government of Canada confirmed on March 13 that some of its servers were breached by attackers making use of the Apache Struts flaw, also identified as CVE-2017-5638.

While the public disclosure for the Apache Struts flaw came on Monday March 6, Canadian Federal IT security administrators apparently weren't aware of the issue until late on Wednesday March 8. The admission came in an Ottawa briefing to Canadian media agencies on March 13.

The Government of Canada took multiple sites down on March 9 including Statistics Canada as well as the

Team of hackers take remote control of Tesla Model S from 12 miles away

Chinese researchers were able to interfere with the car's brakes, door locks and other electronic features, demonstrating an attack that could cause havoc

Now that cars such as Tesla's are increasingly high-tech and connected to the internet, cybersecurity has become as big an issue as traditional safety features. Photograph: Jim Dyson/Getty Images

Three months since the first fatal crash involving a Tesla driving in autopilot mode, hackers have taken remote control of a Tesla Model S from a distance of 12

TECHNOLOGY NEWS | Tue Feb 24, 2015 | 7:58pm EST

Anthem says hack may affect more than 8.8 million other BCBS members

The office building of health insurer Anthem is seen in Los Angeles, California February 5, 2015. REUTERS/Gus Ruelas

Airbnb – Ruby on Rails String Interpolation led to Remote Code Execution

Author: Brett Buerhaus

March 13, 2017 bbuerhaus airbnb, hackerone, rails, RCE, ruby

Authors:

- Ben Sadeghipour
- Brett Buerhaus

@nahamsec and I discovered a Cross-Site Scripting vulnerability a few months ago related to Rails typecasting request variables into JSON. This caused the output to be JSON formatted and the JSON indexes would avoid

Cloudbleed: Cloudflare leaks sensitive data, many major websites affected

By Mihăiță Bamburic | Published 3 weeks ago | Follow @NoFanboi

15 Comments | Like 20 | Share 6 | G+1 4 | Tweet

RCESECURITY
SECURING YOUR WORLD

Vulnerabilities Security S

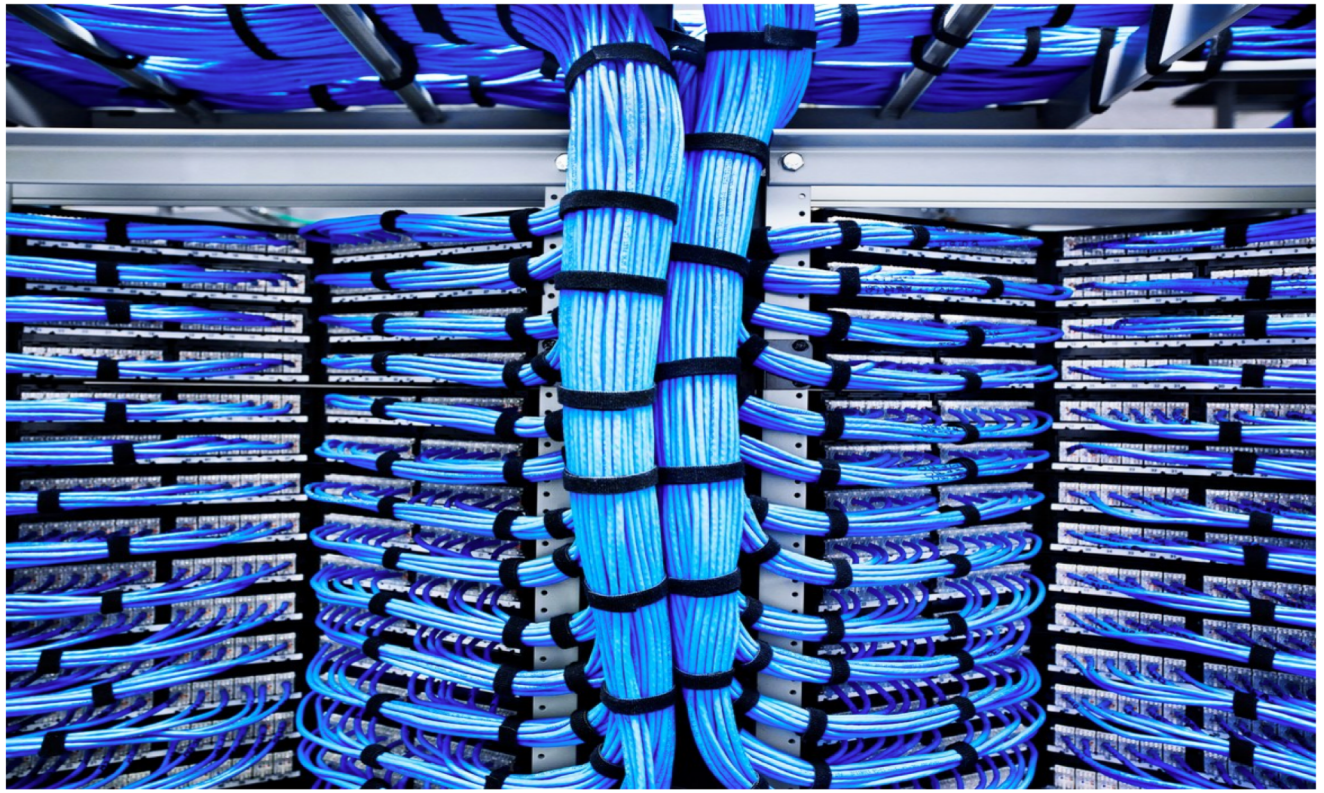
Ok Google, Give Me All Your Internal DNS Information!

In late January, I have found and reported a Server-Side Request Forgery (SSRF) vulnerability on toolbox.googleapps.com to Google's VRP, which could be used to discover and query internal Google DNS servers to extract all kinds of corporate information like used internal IP addresses across the company as well as A and NS records exposing all kinds of hosts like Google's Active Directory structures and also a fancy Minecraft server! 🙂 So here's a quick write-up about the vulnerability.

As you might already know, the G-Suite Toolbox can be used to perform all kinds of trouble-shootings. Among all the available tools, there is one called "Dig" which – on Linux – can be used to query a DNS server for its records of a given domain, just like A- or MX records. In this case Google implemented a nice web interface for that tool to visually lookup DNS information. While it looks like a useful tool to query DNS servers from a Google perspective…

LILY HAY NEWMAN SECURITY 10.21.16 1:04 PM

WHAT WE KNOW ABOUT FRIDAY'S MASSIVE EAST COAST INTERNET OUTAGE



Mozilla wants woeful WoSign certs off the list

Backdating SHA-1 certs is just not on

27 Sep 2016 at 03:58, Richard Chirgwin

Mozilla wants to kick Chinese certificate authority (CA) WoSign out of its trust program.

Security

Chinese CA hands guy base certificates for GitHub, Florida uni

Man-in-the-middle diddle

29 Aug 2016 at 07:57, Darren Pauli




A Chinese certificate authority handed out a base certificate for GitHub and the Univerisity of Central Florida to a mere user in a significant security blunder.

British Mozilla programmer Gervase Markham reported the incident on the browser baron's mailing list saying it occurred more than a year ago in July 2015 but went unreported.

The gaffe meant university sysadmin Stephen Schrauger was handed a certificate for the GitHub domain from issuer WoSign.

143 million compromised Social Security numbers: everything you need to know about the Equifax hack

Equifax confirms Apache Struts security flaw it failed to patch is to blame for hack

The company said the March vulnerability was exploited by hackers.

By Zack Whittaker for Zero Day | September 14, 2017 -- 01:27 GMT (18:27 PDT) | Topic: Security

Equifax Hackers Exploited Months-Old Flaw

by BEN POPKEN

Equifax announced late Wednesday that the source of the hole in its defenses that enabled hackers to plunder its databases was a massive server bug first

Under Armour Says 150 Million MyFitnessPal Accounts Hacked

By **Nick Turner**

March 29, 2018, 4:44 PM EDT *Updated on* March 29, 2018, 6:09 PM EDT

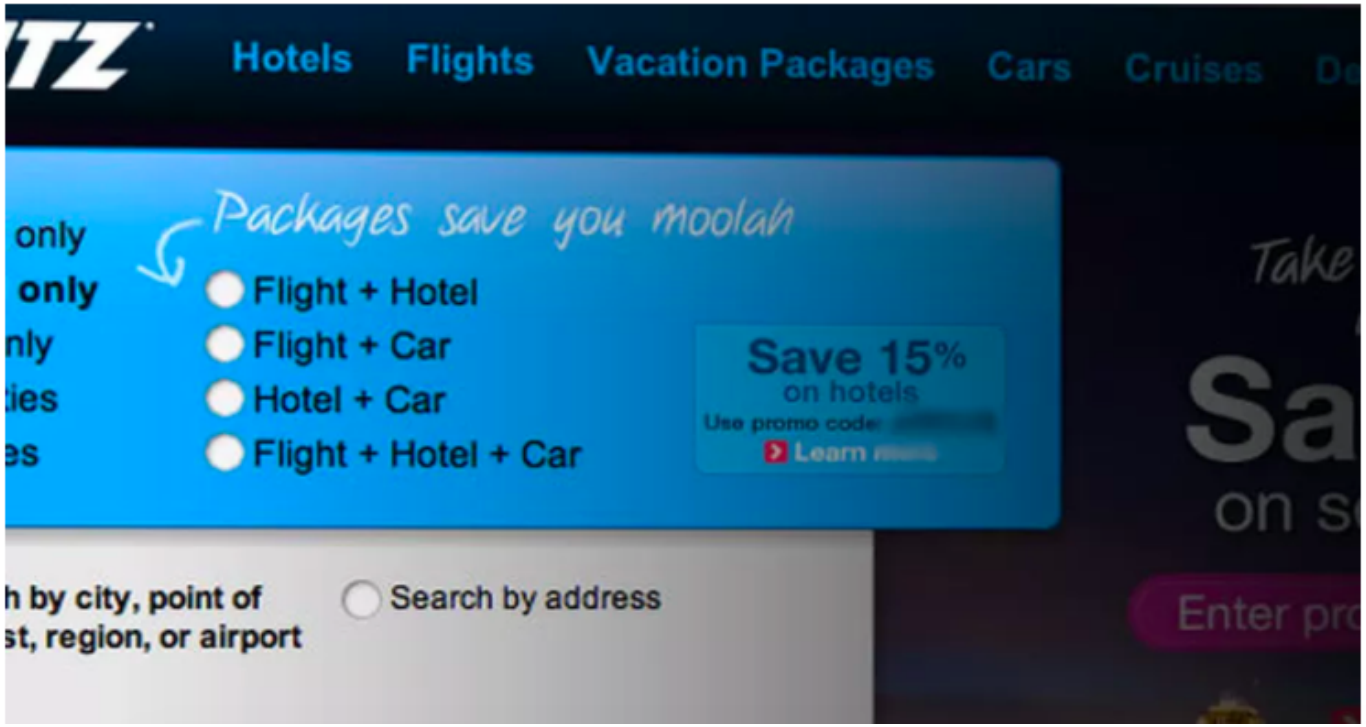
- User names, email accounts and password data were taken
- Shares of the company slide in the wake of the announcement

Under Armour Inc., joining a growing list of corporate victims of hacker attacks, said about 150 million user accounts tied to its MyFitnessPal nutrition-tracking app were breached earlier this year.

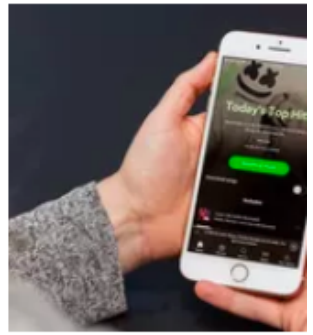
Orbitz says a possible data breach has affected 880,000 credit cards

By Dani Deahl | @danideahl | Mar 20, 2018, 4:28pm EDT

SHARE



MOST RE



Now you can get Spoti
together for just $12.9

New Study Shows 20% of Public AWS S3 Buckets are Writable

BEN LAYER Follow @benlayer

MAR 1, 2018 | CLOUD

Data exposure reports have reached a dizzying pace in the past few months, and the security community has been focused on the risk from multiple angles. Now, a new study from HTTPCS gives us new insight into rates of vulnerable S3 configurations.

SECURITY

Accenture Left AWS S3 Buckets Containing Cloud Credentials Open to Public

he unsecured AWS S3 buckets revealed “significant internal Accenture data, including cloud platform credentials and configurations.”

15 FEB 2018 NEWS

FedEx S3 Bucket Exposes Private Details on Thousands Worldwide

Tara Seals US/North America News Reporter, Infosecurity Magazine

Email Tara

Personal information for thousands of FedEx customers worldwide has been exposed after a legacy Amazon Web Services (AWS) cloud storage server was left open to public access without a password.

Kromtech Security Center researchers stumbled upon the AWS S3 bucket, finding that it contained more than 119,000 scanned documents, including passports, drivers' licenses and Applications for Delivery of Mail Through Agent forms, which contain names, home addresses, phone numbers and ZIP codes.

The victims include citizens of countries around the globe, including Australia, Canada, China, EU countries, Japan, Kuwait, Malaysia, Mexico, Saudi Arabia and others.

Why Not Watch?

18 MAY 2017

How to Get Smart About Prevention

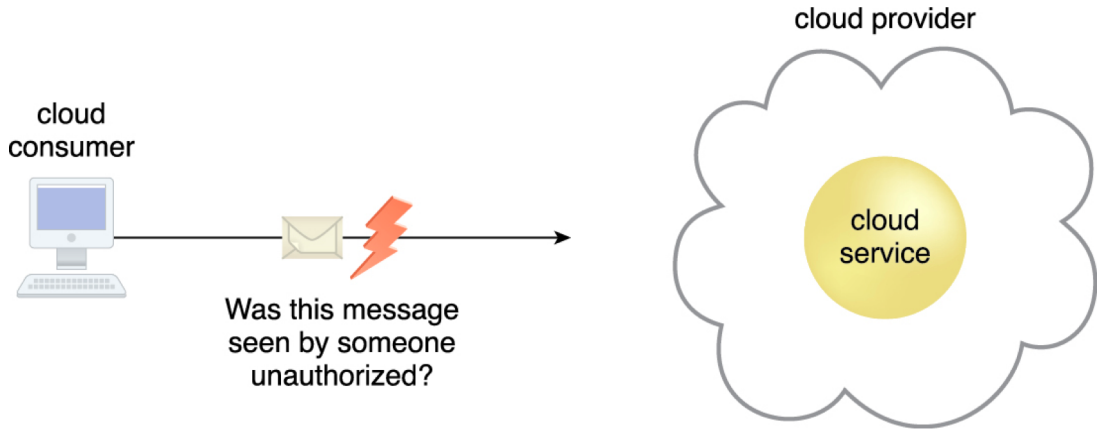
14 SEP 2017

Be Less Man vs Machine, More Ma and Machine

Basic Terms and Concepts

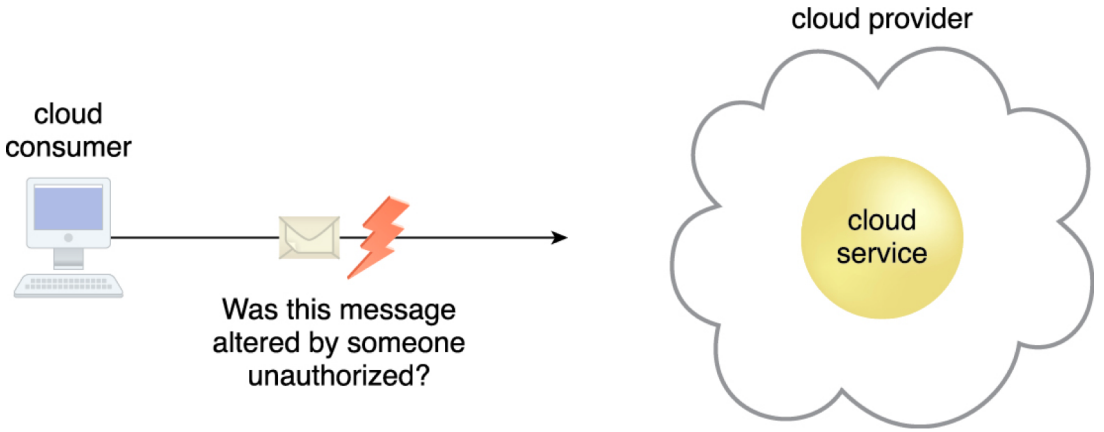
Confidentiality

Only authorized party can get access to data.



Integrity

Guarantee the user that the data it transmitted matches the data received by the cloud service.



Authenticity

Is the data coming from an authorized source?

Availability

Service should be accessible and usable during a specified time period. For a cloud service, the time period is 24/7/365.

Threat

In computer security a threat is a possible danger that might exploit a vulnerability to breach security and therefore cause possible harm.

Vulnerability

A vulnerability is a weakness that can be exploited.

Attack Vectors

- An attack vector is a path or means by which a hacker can gain access to a computer or network server in order to deliver a payload or malicious outcome.
- Attack vectors enable hackers to exploit system vulnerabilities, including the human element.

Risk

Risk is the possibility of loss or harm arising from performing an activity

Security Controls

Security controls are countermeasures used to prevent or respond to security threats and to reduce or avoid risk.

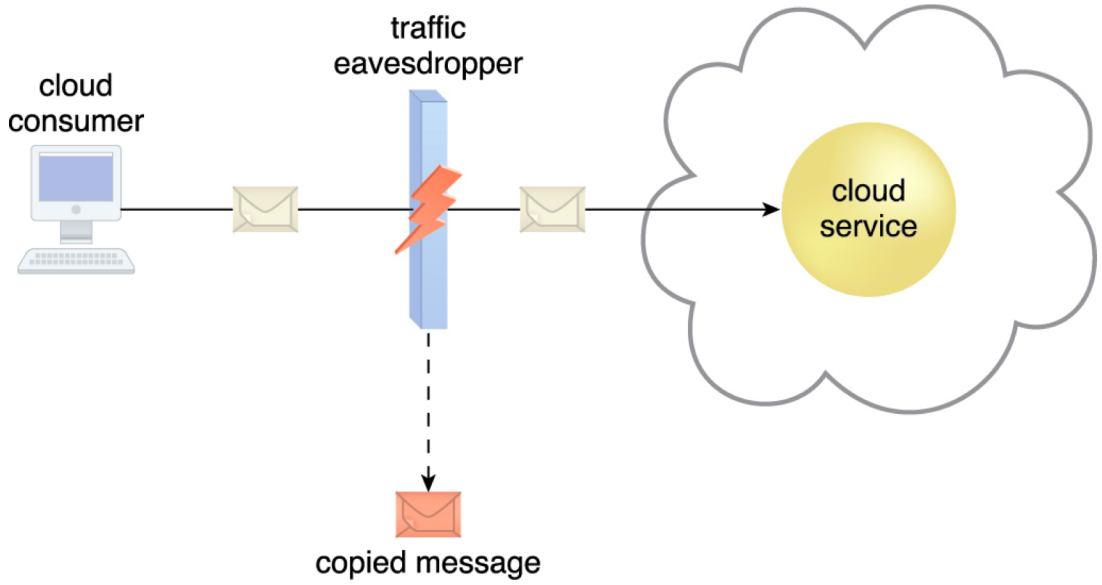
Threat Agent

A threat agent is an entity that poses a threat because it is capable of carrying out an attack.

Cloud Security Threats

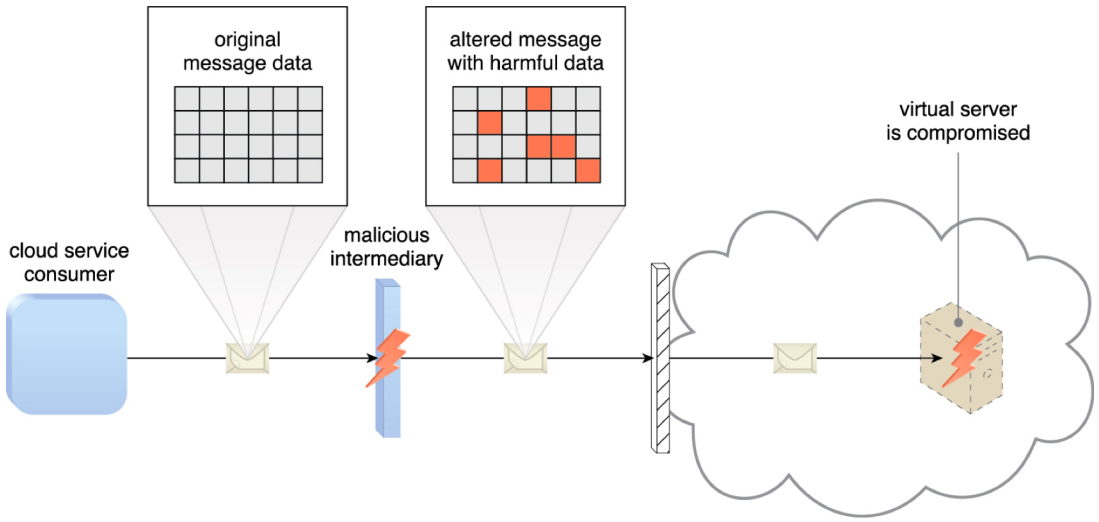
Traffic Eavesdropping

- Traffic eavesdropping occurs when data being transferred to or within a cloud service is passively intercepted for illegitimate information gathering purposes
- This attack's aim is to directly compromise the confidentiality of the data and, possibly, the confidentiality of the relationship between the cloud consumer and cloud provider.
- Because of the passive nature of the attack, it can more easily go undetected for extended periods of time.



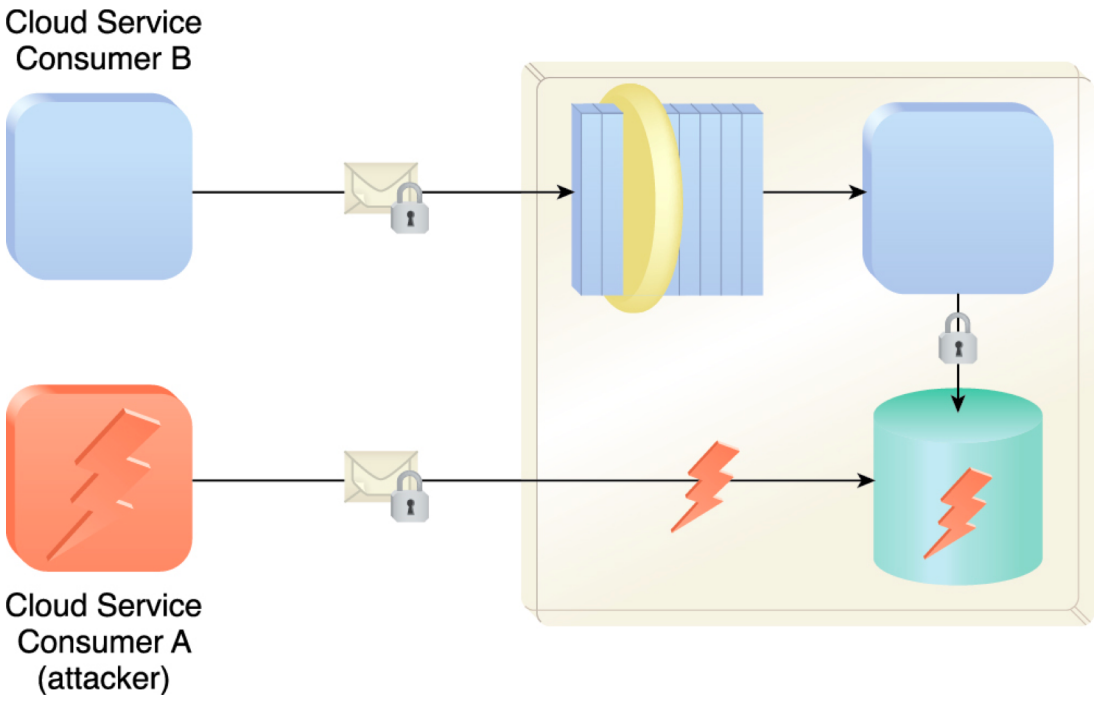
Malicious Intermediary

- The malicious intermediary threat arises when messages are intercepted and altered by a malicious service agent, thereby potentially compromising the message's confidentiality and/ or integrity.
- It may also insert harmful data into the message before forwarding it to its destination.



Insufficient Authorization

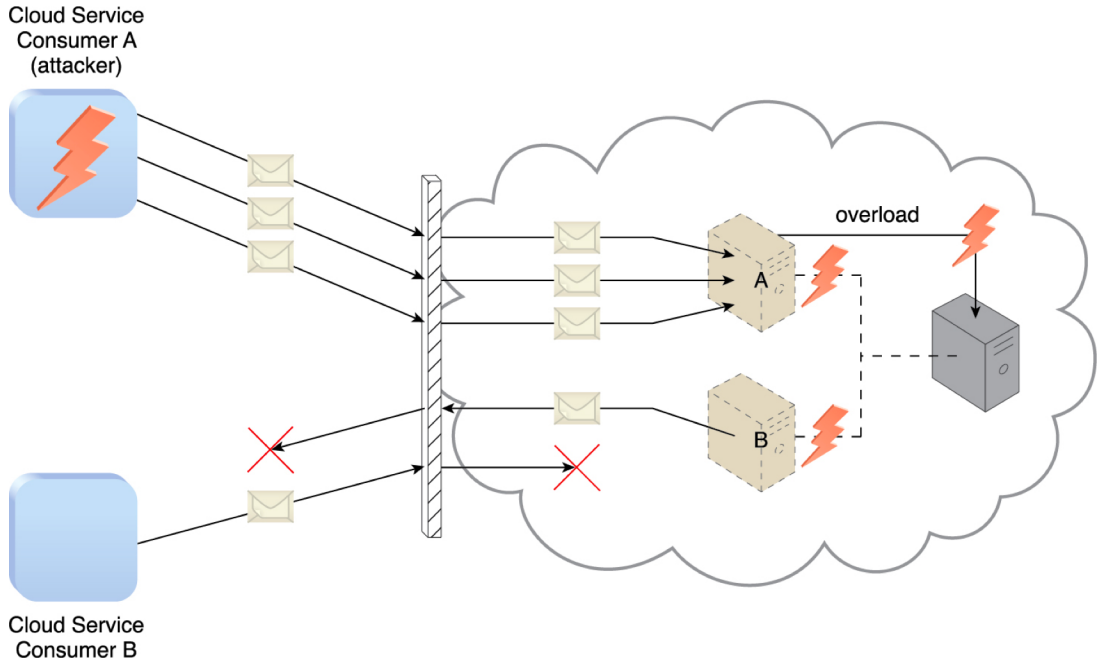
- The insufficient authorization attack occurs when access is granted to an attacker erroneously or too broadly, resulting in the attacker getting access to IT resources that are normally protected.
- This is often a result of the attacker gaining direct access to IT resources that were implemented under the assumption that they would only be accessed by trusted consumer programs.



Denial of Service

The objective of the denial of service (DoS) attack is to overload resources to the point where they cannot function properly. This form of attack is commonly launched in one of the following ways:

- The workload is artificially increased with imitation messages or repeated communication requests.
- The network is overloaded with traffic to reduce its responsiveness and cripple its performance.
- Multiple requests are sent, each of which is designed to consume excessive memory and processing resources. Successful DoS attacks produce server degradation and/ or failure.



Flawed Implementations

- The substandard design, implementation, or configuration of service and it's deployments can have undesirable consequences, beyond runtime exceptions and failures.
- If the cloud provider's software and/ or hardware have inherent security flaws or operational weaknesses, attackers can exploit these vulnerabilities to impair the integrity, confidentiality, and/ or availability of cloud provider IT resources and cloud consumer IT resources hosted by the cloud provider.

Cloud Security Mechanisms

Encryption

- Data, by default, is coded in a readable format known as plaintext. When transmitted over a network, plaintext is vulnerable to unauthorized and potentially malicious access.
- The encryption mechanism is a digital coding system dedicated to preserving the confidentiality and integrity of data. It is used for encoding plaintext data into a protected and unreadable format.
- Encryption can help counter the traffic eavesdropping, malicious intermediary, insufficient authorization, and overlapping trust boundaries security threats.
- There are two common forms of encryption known as **symmetric** encryption and **asymmetric** encryption.

Symmetric Encryption

- Symmetric encryption uses the same key for both encryption and decryption, both of which are performed by authorized parties that use the one shared key.
- Also known as secret key cryptography, messages that are encrypted with a specific key can be decrypted by only that same key.
- Parties that rightfully decrypt the data are provided with evidence that the original encryption was performed by parties that rightfully possess the key.
- A basic authentication check is always performed, because only authorized parties that own the key can create messages. This maintains and verifies data confidentiality.
- Note that symmetrical encryption does not have the characteristic of non-repudiation, since determining exactly which party performed the message encryption or decryption is not possible if more than one party is in possession of the key.

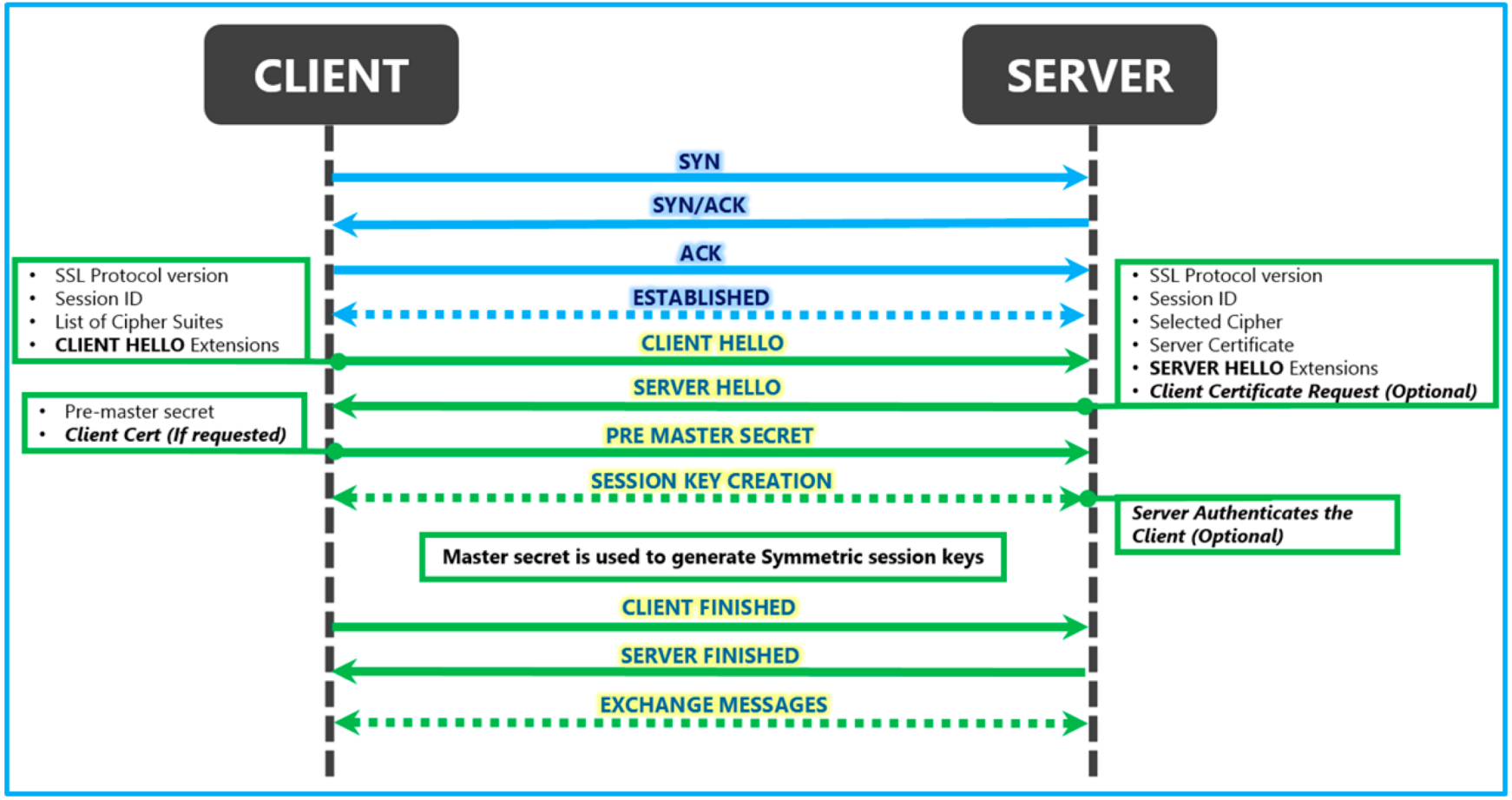
Asymmetric Encryption

- Asymmetric encryption relies on the use of two different keys, namely a private key and a public key.
- With asymmetric encryption (which is also referred to as public key cryptography), the private key is known only to its owner while the public key is commonly available.
- A document that was encrypted with a private key can only be correctly decrypted with the corresponding public key.
- Conversely, a document that was encrypted with a public key can be decrypted only using its private key counterpart.
- As a result of two different keys being used instead of just the one, asymmetric encryption is almost always computationally slower than symmetric encryption.

Securing Web-based Data Transmission

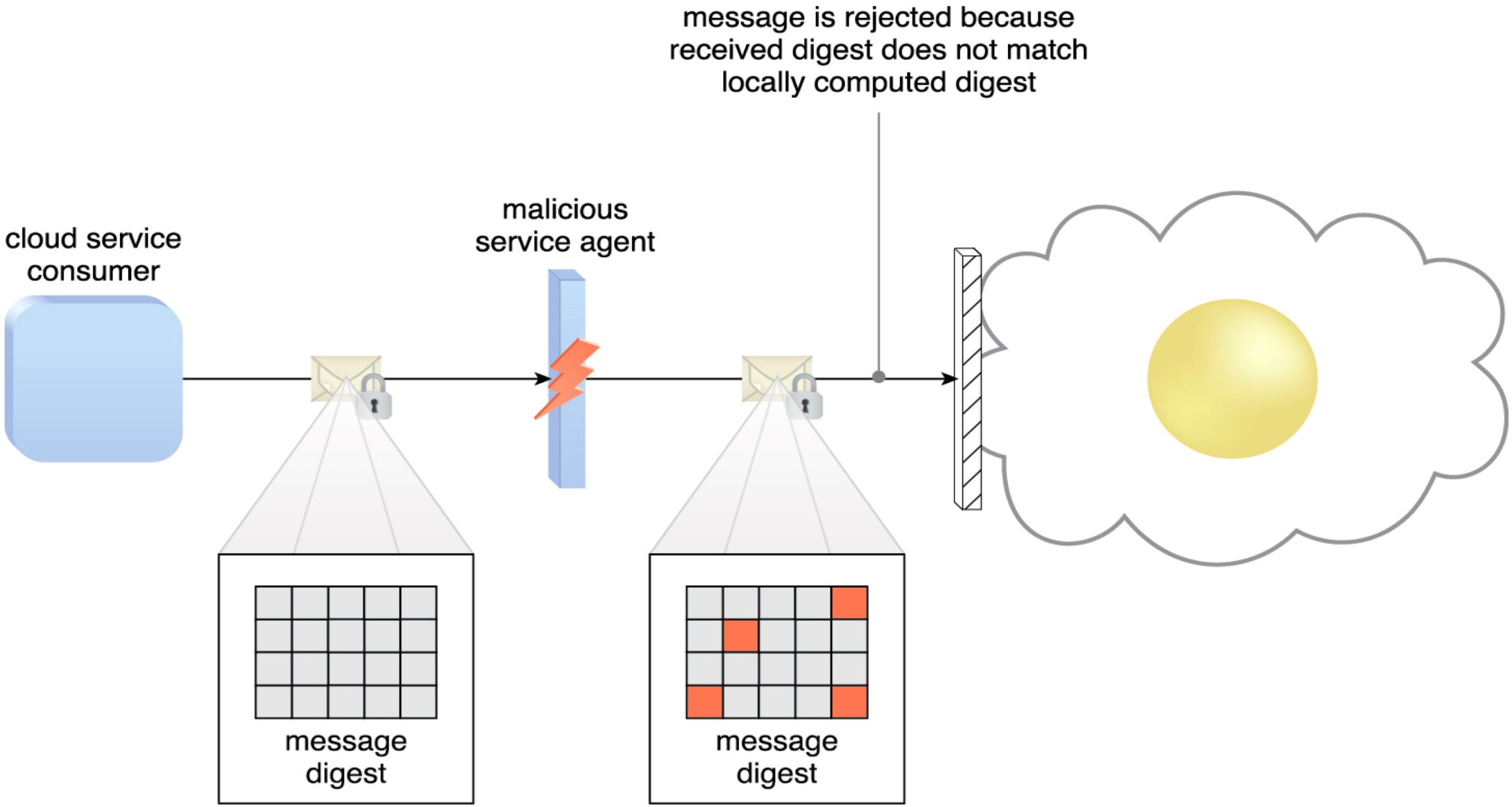
- The encryption mechanism, when used to secure Web-based data transmissions, is most commonly applied via HTTPS, which refers to the use of SSL/ TLS as an underlying encryption protocol for HTTP. TLS (transport layer security) is the successor to the SSL (secure sockets layer) technology.
- Because asymmetric encryption is usually more time-consuming than symmetric encryption, TLS uses the former only for its key exchange method.
- TLS systems then switch to symmetric encryption once the keys have been exchanged.
- Most TLS implementations primarily support RSA as the chief asymmetrical encryption cipher, while ciphers such as Triple-DES, and AES are supported for symmetrical encryption.

SSL Handshake



Hashing

- The hashing mechanism is used when a one-way, non-reversible form of data protection is required.
- Hashing can be used to derive a message digest from a message, which is often of a fixed length and smaller than the original message.
- The message sender can then utilize the hashing mechanism to attach the message digest to the message.
- The recipient applies the same hash function to the message to verify that the produced message digest is identical to the one that accompanied the message.
- Any alteration to the original data results in an entirely different message digest and clearly indicates that tampering has occurred.


message is rejected because
received digest does not match
locally computed digest
cloud service
consumer
malicious
service agent
message
digest
message
digest

Digital Signature

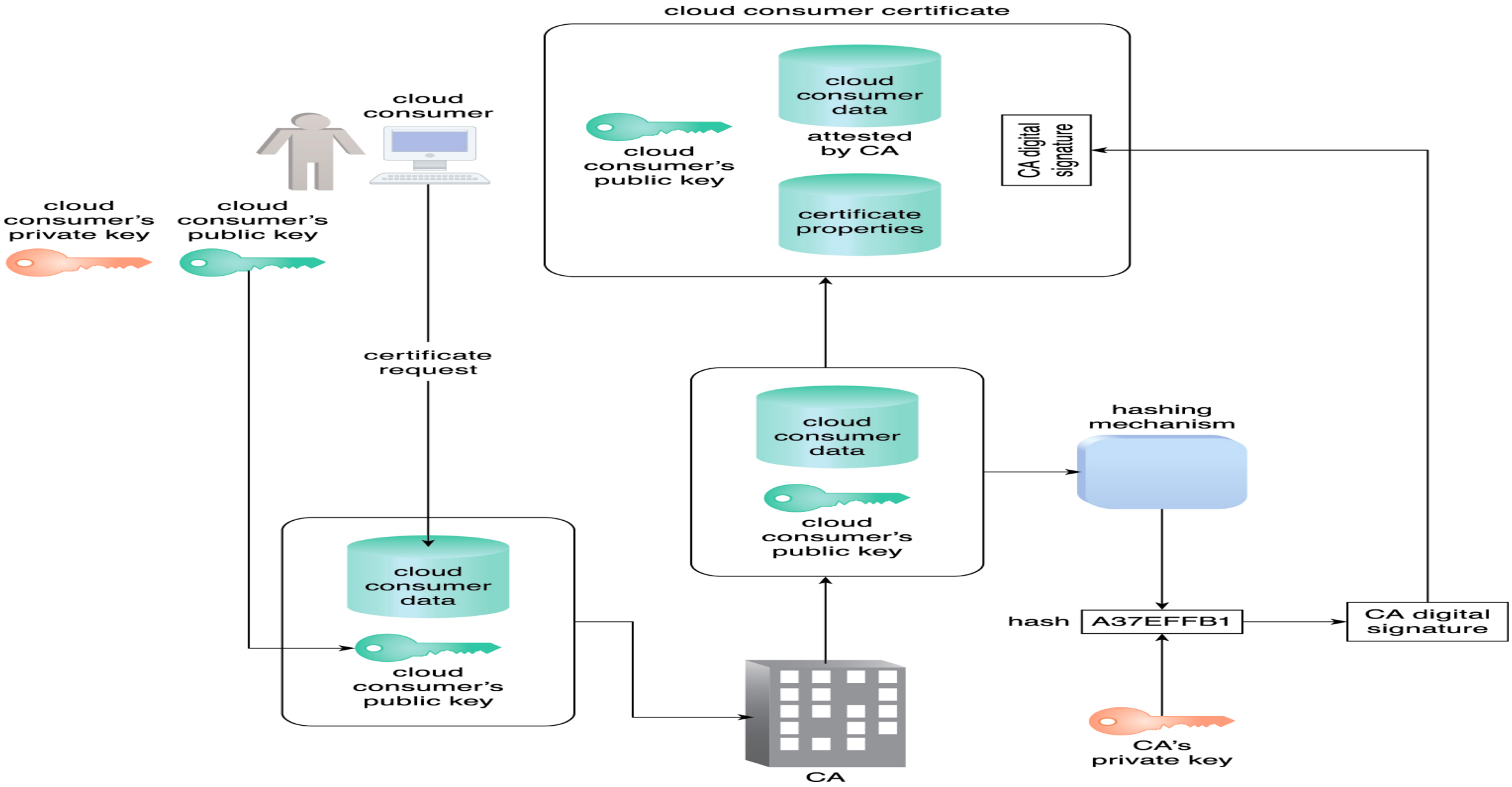
- The digital signature mechanism is a means of providing data authenticity and integrity through authentication and non-repudiation.
- A message is assigned a digital signature prior to transmission, which is then rendered invalid if the message experiences any subsequent, unauthorized modifications.
- A digital signature provides evidence that the message received is the same as the one created by its rightful sender.
- Both hashing and asymmetrical encryption are involved in the creation of a digital signature, which essentially exists as a message digest that was encrypted by a private key and appended to the original message.
- The recipient verifies the signature validity and uses the corresponding public key to decrypt the digital signature, which produces the message digest.
- The hashing mechanism can also be applied to the original message to produce this message digest. Identical results from the two different processes indicate that the message maintained its integrity.

Public Key Infrastructure (PKI)

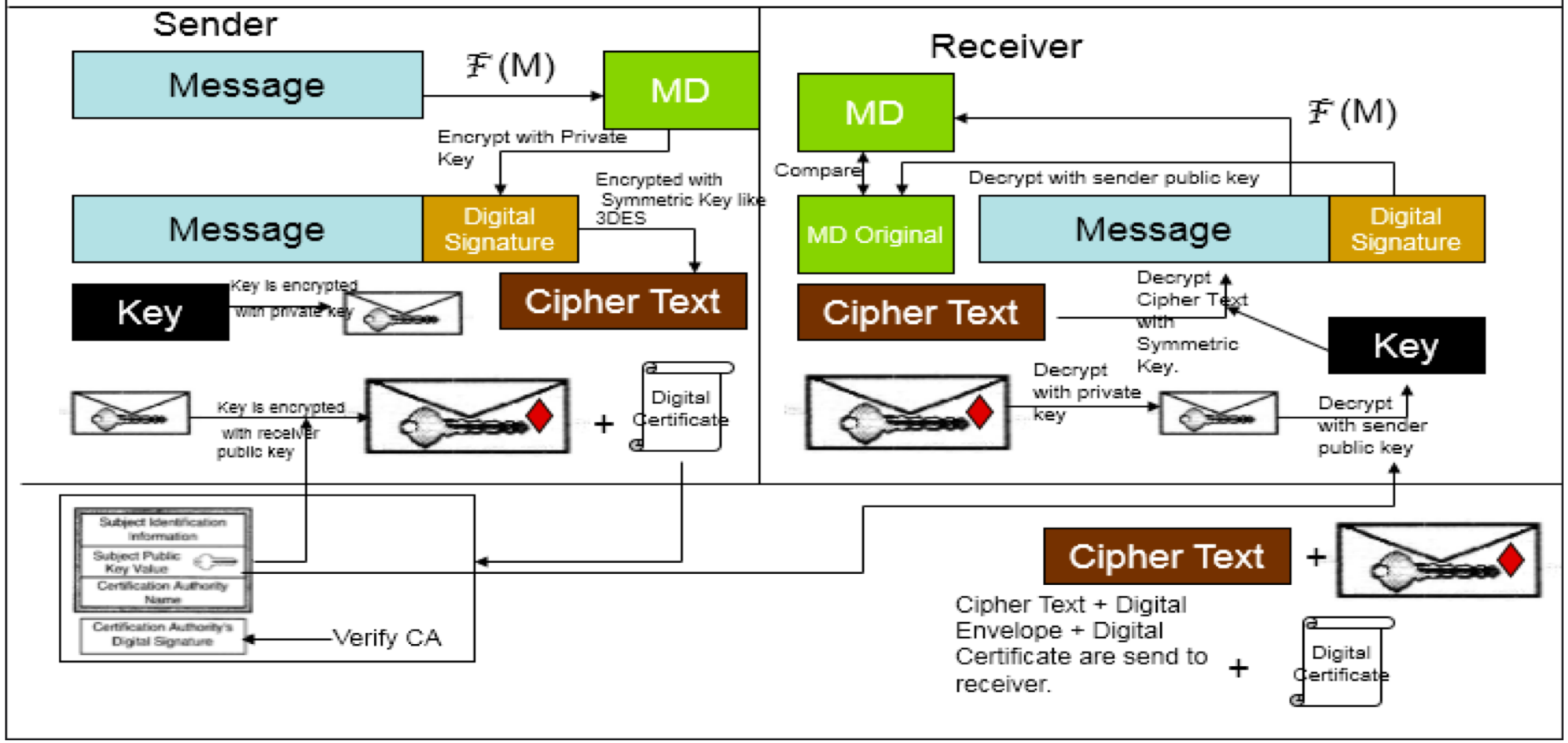
- A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption.
- The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email.
- It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

Public Key Infrastructure (PKI)

- In cryptography, a PKI is an arrangement that binds public keys with respective identities of entities (like persons and organizations).
- The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA).
- Depending on the assurance level of the binding, this may be carried out by an automated process or under human supervision.


cloud consumer certificate
cloud consumer data
attested by CA
cloud consumer's public key
certificate properties
CA digital signature
cloud consumer
cloud consumer's private key
cloud consumer's public key
certificate request
cloud consumer data
cloud consumer's public key
hashing mechanism
cloud consumer data
cloud consumer's public key
hash
A37EFFB1
CA digital signature
CA's private key
CA

How does PKI Work?



Hardened Server Images

- Hardening is the process of stripping unnecessary software from a system to limit potential vulnerabilities that can be exploited by attackers.
- Removing redundant programs, closing unnecessary server ports, and disabling unused services, internal root accounts, and guest access are all examples of hardening.

Application Security

The Core Security Problem: Users Can Submit Arbitrary Input

The application must assume that all input is potentially malicious, and must take steps to ensure that attackers cannot use crafted input to compromise the application by interfering with its logic and behavior and gaining unauthorized access to its data and functionality.

User Input

- Users can interfere with any piece of data transmitted between the client and the server, including request parameters, cookies, and HTTP headers.
- Any security controls implemented on the client side, such as input validation checks, can be easily circumvented.
- Users can send requests in any sequence, and can submit parameters at a different stage than the application expects, more than once, or not at all.
- Any assumption which developers make about how users will interact with the application may be violated.
- Users are not restricted to using only a web browser to access the application.
- There are numerous widely available tools that operate alongside, or independently of, a browser, to help attack web applications. These tools can make requests that no browser would ordinarily make, and can generate huge numbers of requests quickly to find and exploit problems.

Attack Vectors

The majority of attacks against web applications involve sending input to the server which is crafted to cause some event that was not expected or desired by the application's designer.

Attack Vector Examples

- Changing the price of a product transmitted in a hidden HTML form field, to fraudulently purchase the product for a cheaper amount.
- Modifying a session token transmitted in an HTTP cookie, to hijack the session of another authenticated user.
- Removing certain parameters that are normally submitted, to exploit a logic flaw in the application's processing.
- Altering some input that will be processed by a back-end database, to inject a malicious database query and so access sensitive data.

Core Defense Mechanisms

Authentication

Authenticating a user involves establishing that the user is in fact who he claims to be.

Session Management

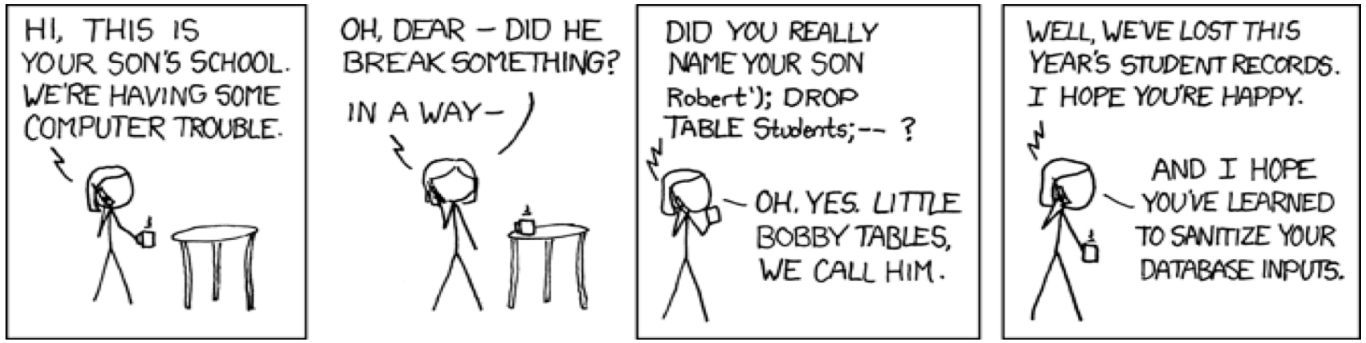
- After successfully logging in to the application, the user will access various pages and functions, making a series of HTTP requests from their browser.
- At the same time, the application will be receiving countless other requests from different users, some of whom are authenticated and some of whom are anonymous.
- In order to enforce effective access control, the application needs a way of identifying and processing the series of requests that originate from each unique user.
- Virtually all web applications meet this requirement by creating a session for each user and issuing the user a token that identifies the session.
- In terms of attack surface, the session management mechanism is highly dependent on the security of its tokens, and the majority of attacks against it seek to compromise the tokens issued to other users.

Access Control

- The access control mechanism usually needs to implement some fine-grained logic, with different considerations being relevant to different areas of the application and different types of functionality.
- An application might support numerous different user roles, each involving different combinations of specific privileges.
- Individual users may be permitted to access a subset of the total data held within the application.
- Because of the complex nature of typical access control requirements, this mechanism is a frequent source of security vulnerabilities that enable an attacker to gain unauthorized access to data and functionality.

Handling User Input

- Varieties of Input
- Input Sanitization
- Safe Data Handling



Handling Hackers

Mapping the Application

- Discovering hidden content
- Discovering hidden parameters
- Identifying entry points for user input
- Identifying server-side technologies
- Identifying server-side functionality

Bypassing Client Side Controls

- Hidden form fields
- URL Parameters
- Referrer Header
- Data length input
- Script based validation

Attacking Authentication

- Bad passwords
- Brute-Forcible Login
- Verbose Failure Messages
- Password Change Functionality
- “Remember Me” Functionality
- Storing credentials in plain-text in database
- No “salt” used

Code Injection into Interpreted Languages

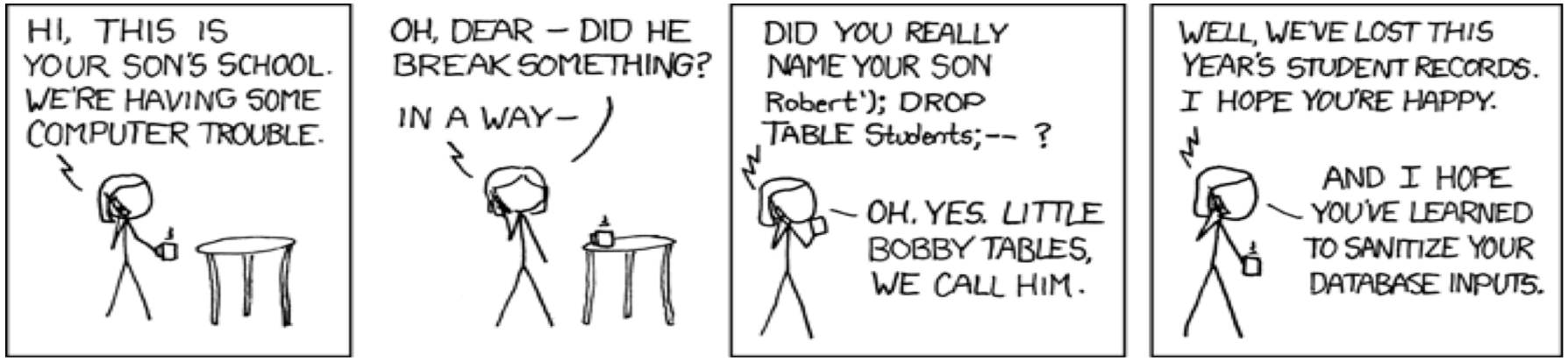
- An interpreted language is one whose execution involves a runtime component that interprets the code of the language and carries out the instructions that it contains.
- Because of the way that interpreted languages are executed, there arises a family of vulnerabilities known as code injection.
- In any useful application, user-supplied data will be received, manipulated, and acted upon.
- The code that is processed by the interpreter will, therefore, comprise a mix of the instructions written by the programmer and the data supplied by the user.

Code Injection Example

codeInjection.sh
```
#!/bin/bash
echo $1
```
./codeInjection.sh “`ls -la`”

Injecting Code in SQL

- SQL injection is the elder statesman of code injection attacks, being still one of the more prevalent vulnerabilities in the wild, and frequently one of the most devastating.
- SQL injection can enable an anonymous attacker to read and modify all data stored within the database, and even take full control of the server on which the database is running.



SQL Injection Example

The following line of code illustrates this vulnerability:

```
statement = " SELECT * FROM users WHERE name = ' " + userName + " ; "
```

This SQL code is designed to pull up the records of the specified username from its table of users. However, if the "userName" variable is crafted in a specific way by a malicious user, the SQL statement may do more than the code author intended. For example, setting the "userName" variable as:

```
' OR '1'='1
```

or using comments to even block the rest of the query (there are three types of SQL comments[13]). All three lines have a space at the end:

```
' OR '1'='1' --
' OR '1'='1' ({
' OR '1'='1' /*
```

renders one of the following SQL statements by the parent language:

```
SELECT * FROM users WHERE name = '' OR '1'='1';
```

```
SELECT * FROM users WHERE name = '' OR '1'='1' -- ';
```

Cross-site scripting

- Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications.
- XSS enables attackers to inject client-side scripts into web pages viewed by other users.
- A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

http://bobssite.org?q=<script%20type='text/javascript'>alert('xss');</script>

XSS (Cross Site Scripting) Prevention Cheat Sheet

https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

Cross-site request forgery

- Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.
- CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.
- With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing.
- If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth.
- If the victim is an administrative account, CSRF can compromise the entire web application.

Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet

https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet

OWASP recommends two separate checks to protect against CSRF.

1. Check standard headers to verify the request is same origin
2. Check CSRF token

Additional Resources

https://spring2018.csye6225.com/