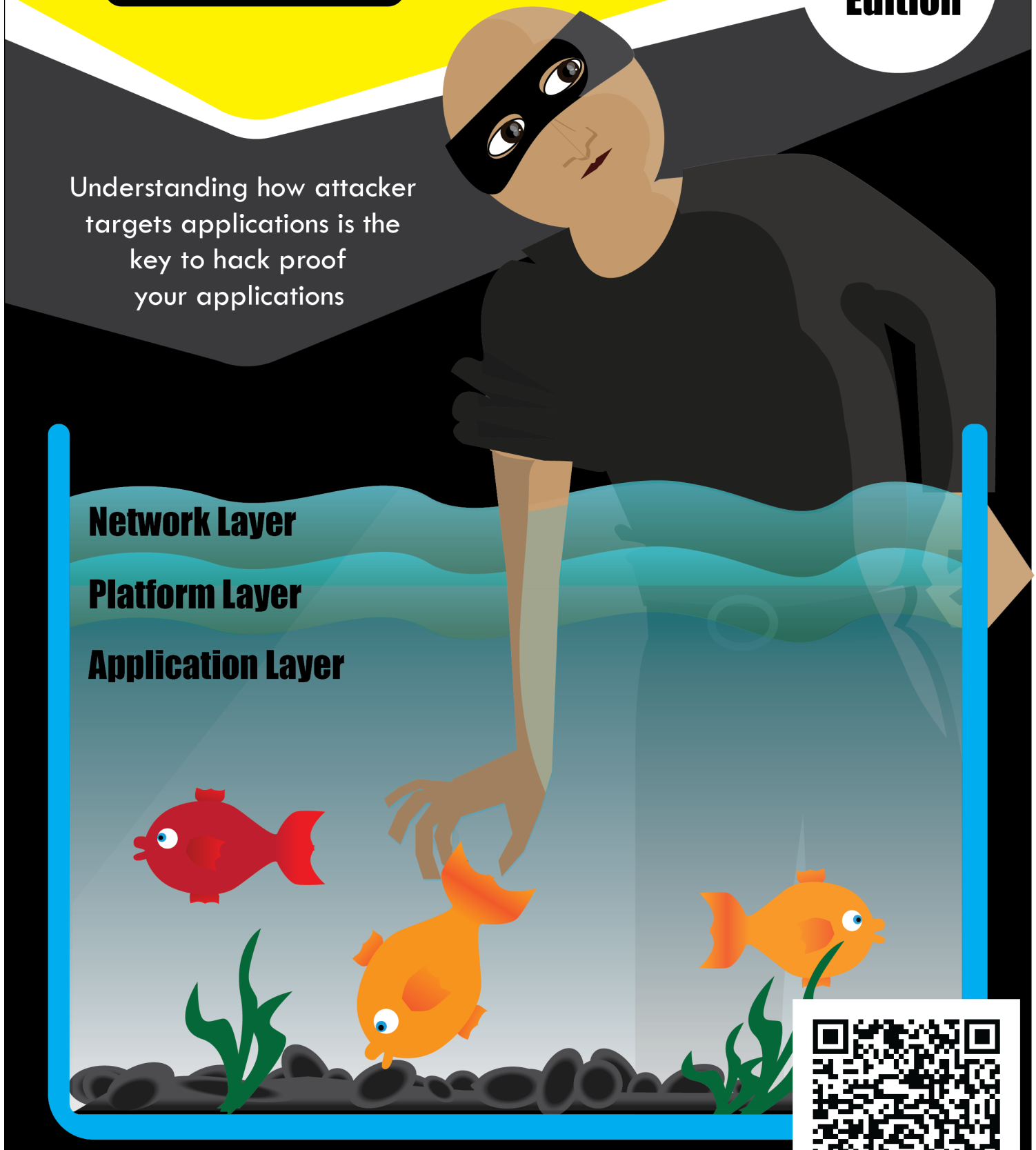
Web Application Security

The Fast Guide

The 1st Edition

Understanding how attacker targets applications is the key to hack proof your applications



BY SAMI KHIAMI

Web Application Security

The Fast Guide

By

Dr.Sami Khiami

(2017)

Table of contents

Chapter 1

information

Security overview

1.1 Information security definition

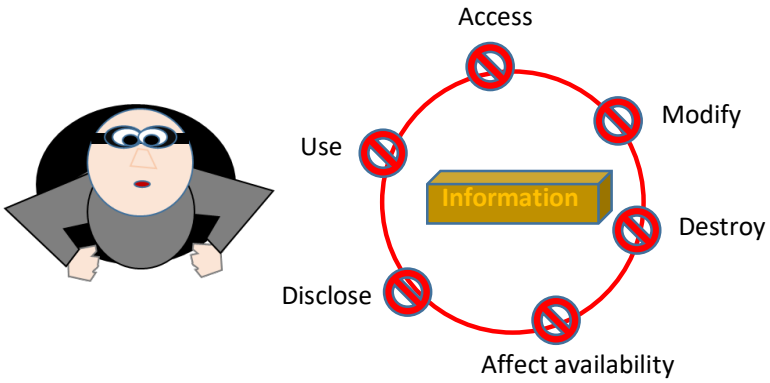


Figure 1 :main threats affecting applications

Information is like any other asset subject to unintended or malicious activities that might affect its confidentiality, integrity or availability hence a defensive practice, activities should take place to help protecting these precious assets.
Other definitions might concentrate more on safeguarding information in its different status such as static stored in databases, files or dynamic moving over different carriers or while it is Processed.

1.2 Applying security

1.2.1 Design & Build it to be secure:

this approach might depend on building the application over a framework with security focus where security becomes part of application itself with minimum risk of security vulnerabilities.
Sometimes this approach is reached through a special process like development methodology or as programming language that enforce security.
This approach might look perfect for new applications but when it comes to old or legacy application this becomes nonrealistic approach.

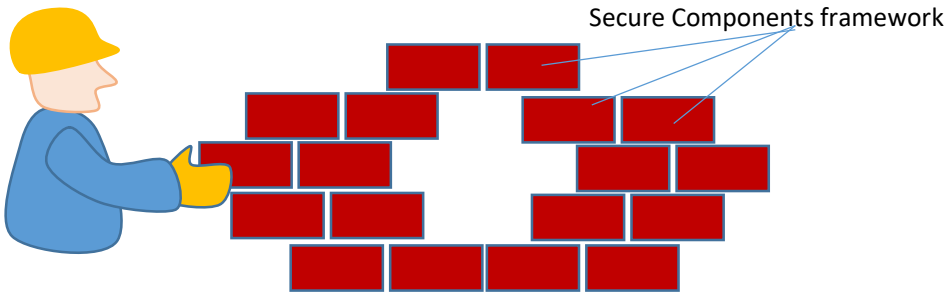


Figure 2:Design & Build it to be secured

1.2.2 Verify it is secure:

This approach depends on vulnerability analysis by investigating different vulnerabilities to be sure that main and known ones are covered.
The next step to apply security through that approach is to reinforce and fix vulnerabilities.
This approach can be usefull in new systems and legacy ones.

- Vulnerability analysis can be done through **application** or even **manually** depending on the analyzed vulnerability.
- Vulnerability analysis can be done using :
  - **static** methods like auditing the application source code
  - **Dynamic** method: the analysis is done in the run time by observing the behavior of the system.

  Using the static method might give the maximum coverage for most existing vulnerabilities but it might have issues of false alerts in time when the dynamic method we can be sure of correctness but no guarantee for complete coverage of vulnerabilities.

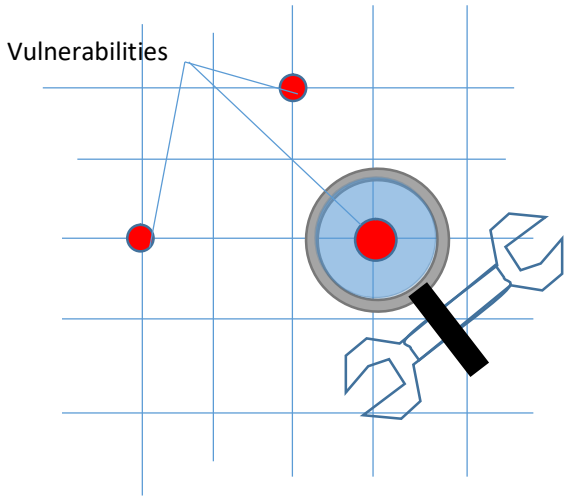


Figure 3: security by verification (analyze, Identify and fix)

1.2.3 Protect it:

This approach depends on building a run time environment that will help in protecting the application vulnerability from being exploited this approach can be applied through two methods:

1- Proxy approach that will isolate and detach application from other components in the system which minimize the ability to exploit the vulnerabilities.
2- Embed monitoring capabilities in infrastructure components (Browser, language runtime) to enable monitoring behavior, isolate and quarantine any threat.

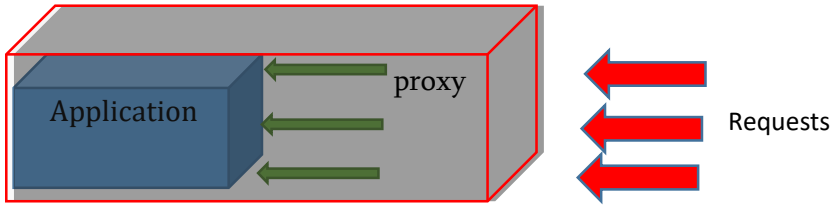


Figure 4: isolate the application using proxy

Even though the presented approaches are categorized in different classes but a hybrid use can be applied sometimes depending of the nature of application.

1.3 Layered Security

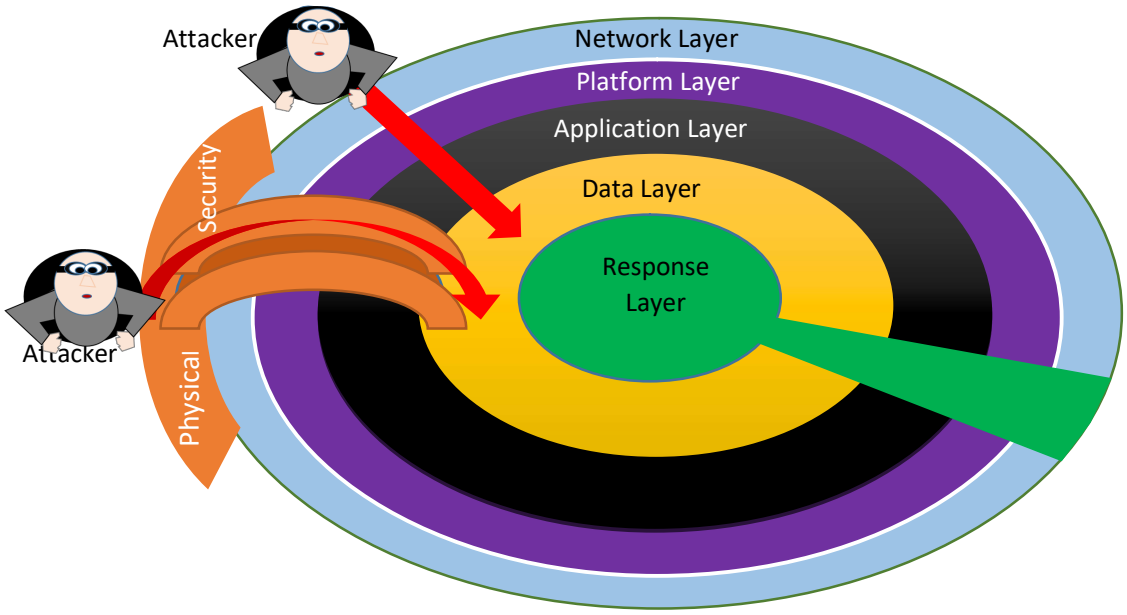


Figure 5: layers based security

One of the most efficient ways to deal with security issues in general and information security in specific is to apply a layered based model in order to be able to understand threats and apply necessary countermeasures for it.

What makes this model suitable for security is the architecture of network and information systems nowadays where most of the interactions are between users and information systems through the network as a set of requests sent from the beneficiary to the server that will handle the request, process any sent information, retrieve or manipulate data.in that context the data become the core of model as it is the main important asset that need to be protected.

Many models were created to embody the layered security approach from different perspectives.

Some models took in consideration the security policy and user dimension and other focus more on the main layers:

1.3.1 The Physical layer:

We mean by the physical layer the direct physical access to hardware. As illustrated in the chart above the access to the physical layer can be very direct and dangerous because attacker can cause direct damage or compromise network, processing, and storage devices. As example causing a denial of service that work on a server is simply doable by unplugging the power cord of that server. This is why physical security of data centers is an issue that needs to be taken seriously.

A well designed architecture should allow response to attack even with physical based attacks as example sending notification or raising an alarm.

1.3.2 Network Layer:

When the attacker doesn't have any direct access to the physical hardware the only available path is through external layers toward the core where the data assets resides.
Compromising network layer will make it easy for attacker to disclose, alter, or make unavailable mainly the data in motion sent by legitimate user or response sent by the server. Network layer in that model represent all activities, devices and protocols used to transfer data from its source to destination.

1.3.3 Platform layer:

The platform layer represents the carrier of application layer it provides the interface between hardware devices and the application layer in addition to process and file management.
This layer is normally reflected through operating system and any used framework or server software that host the application.

1.3.4 Application layer:

This layer represents all input processing, storage, retrieval, manipulation and output activities done on server side or client side. This layer depends on services it gets from the platform layer.

1.3.5 Data layer:

This is the layer where the precious assets reside, as it is known that the Data is the real asset in information systems.

If an attacker is able to reach this layer the information system is considered as compromised.

1.3.6 The response layer:

This layer is the deepest layer it encompasses all Data and system recovery, monitoring, logging and notification activities.

This layer safety is critical because it is the only guarantee that the data will be partially or totally recovered after an attack or at least knowing that the attack took place.
Response layer is an abstract layer because its contents might be distributed over network, platform and application layer

1.4 The security of layers:

in a layer based model each layer provides services to the next layer in order. one of the provided services is security thus each layer is responsible of preventing any malicious attack from passing through to the next layer.but since layers hold different nature it is sometime impossible for a specific layer to stop an attack that ment to target deeper layer.lot of malicious requests can travel freely without any problem through a specific layer as a legitimte requests because request does not contain any sign of malicious activity related to that layer.
Attacker might need to compromise more than one layer to be able to fulfill the attack goals. Compromising a layer is not always the goal of attack it might be only a step to compromise deeper layer to realize the target of attack.
The following drawing illustrates some examples of attack scenarios:

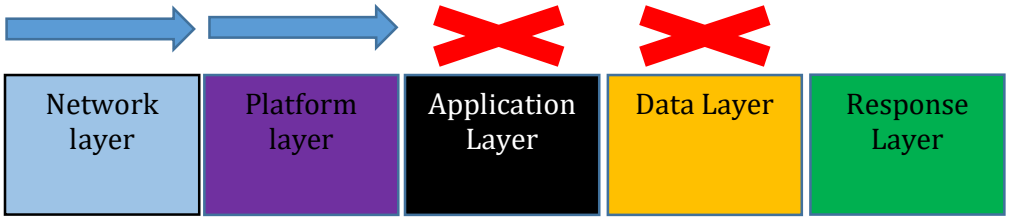


Figure 6: Attacker bypass Network layer, platform layer and compromise Application layer to reach data

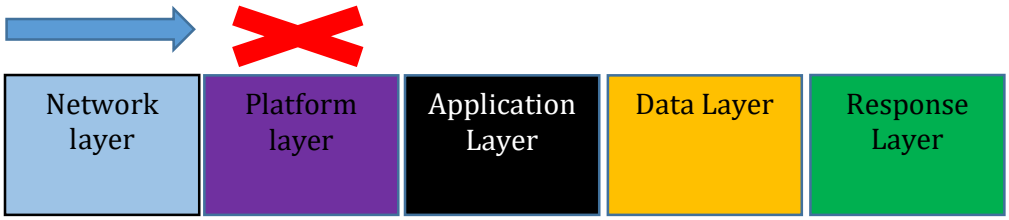


Figure 7: Attacker bypass network layer and compromise platform layer to cause denial of service

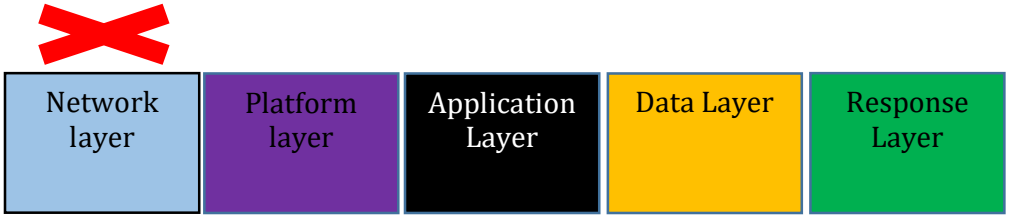


Figure 8: Attacker compromise Network layer and steal data while it is sent by man in the middle attack

It is important to understand that the security is as strong as the weekest layer which means that the compromization of any layer might cause a security breach of the system.

This is why we should defreniciate between various vulnerabilities, attacks, techniques, technologies and tools used to secure each layer.
Our focus in this subject is web application security so we will be concentrating on layers directly related to application namely application layer.

1.5 Application layer security:

Application layer as mentioned is the layer where all the logic of input, processing, manipulation, storage and output reside that makes this layer the place containing the customized component thus the components with less maturity which makes it the most tempting to malicious attacks.

1.6 Defense mechanisms

To be able to defend the application we need to specify the main mechanisms used to make this possible.
This approach emphasizes heavily the application security noting that some other aspects needs to be considered if we target general defense mechanisms
The actual focus is based on the ability to control the access, the attacker and to enable full monitoring capabilities over user input and application:

1.6.1 Access:

this part is about controlling the user privileges in term of access to data and functionality. This target is normally covered in web application by three main mechanisms:

a. **Session management**

Session management is the method in which the server can handle subsequent requests coming from the same user, meaning that it is the way the server differentiates various requests coming from different clients.
Http as a protocol does not provide this service as it is called stateless protocol.
In general, all the application need to provide an approach to help dealing with requested sent by various user keeping track for each unique user.
The common way to allow session management in an application is to create a session structure and generate the session token. The session structure is dedicated to track user interaction through the unique generated token.
Tokens are long, randomly generated strings that are unique for the user. Tokens are transmitted using different methods the most common is HTTP cookies other methods like URL strings or hidden fields can be used too.
Session for specific user is destroyed automatically after a period of time if no interaction between the client and the server is initiated, this period can be set by the application and it is usually about 20 minutes.