

SECURITY TESTING

Insecure Communication

Abstract

Security of an application has always remained a challenge for the application development teams. This white paper stresses the need for effective security testing and filling up gaps because of insecure communication protocols. No matter how well a given system may have been developed, the nature of today's complex systems with large volumes of code, complex internal interactions, interoperability with uncertain external components, unknown interdependencies coupled with vendor cost and schedule pressures, means that exploitable flaws will always be present or surface over time. Accordingly, security testing must fill the gap in system development and actual operation of these systems. Organizations that have an organized, systematic, comprehensive, on-going, and priority driven security testing regimen are in a much better position to make prudent investments to enhance the security posture of their systems.

The Problem

Web applications galore and are difficult to run effectively without the added efficiency and communications brought about by the Internet. At the same time, the Internet has brought about problems as the result of intruder attacks, both manual and automated, which can cost

many organizations excessive amounts of money in damages and lost efficiency.

Thus, organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack. Computer systems today are more powerful and more reliable than in the past; however they are also more difficult to manage.

Understanding Insecure Communication Security Testing

Organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack. Sending information over the internet, whilst preserving its integrity is becoming a must. However information sent over insecure channels makes it vulnerable to sniffing, or manipulation whilst in transit. This leads to majority of web application frameworks vulnerable to insecure communication.

As internet communication is primarily based on TCP/IP protocols (operating at the transport layer of the OSI model), and takes place in plain text, thus making is easily interceptable. To make matters worse, traffic sniffing tools are widely available which facilitate packet sniffing in transit.



The different technologies available to ensure secure communication are:

- Secure Sockets Layer (SSL)
- Secure Electronic Transaction (SET)
- Secure HTTP (HTTPS)
- Secure Shell (SSH)
- IPsec

Secure Sockets Layer (SSL)

SSL is the most widely deployed protocol for securing HTTP communication, and works above the 4th layer of the OSI model. HTTP when deployed over SSL or Transport Layer Security (TLS) is popularly known as HTTPS. Testing for security on most popularly used communication ports such as 443 (HTTPS) and 80 (HTTP) is essential.

SSL ensure data confidentiality and integrity as well as proving server authentication when a client establishes/accepts a connection which is mostly the case in Business-to-Customer transactions i.e., online shopping.

As verification between the server and client takes place during the SSL handshake process, important information about the certificates, security cipher, master and session keys, public or private encryption etc are determined. It is imperative that authentication take place individually both on the client and server end.

Secure Electronic Transaction (SET)

Secure Electronic Transaction (SET) is an open encryption and security specification that is designed for protecting credit card transactions on the Internet. The pioneering work in this area was done in 1996 by MasterCard and Visa jointly. The need for SET came from the fact that MasterCard and Visa realized that for e-commerce payment processing, software vendors were coming up with new and conflicting standards. SET protocol testing can be performed from various perspectives i.e., from the cardholder's perspective, the payment gateway, the merchant, or perhaps even the acquirer.

Secure HTTP (HTTPS)

HTTPS is a protocol to transfer encrypted data over the Internet. HTTPS testing aims to ensure that all relevant aspects of e-commerce hosting (primary use of HTTPS) are reliable and cannot be compromised.

For example, use of dedicated IP address to enable verification of the security certificate. Testing may also encompass to see if all linked URLs include the full server path i.e., https://. For images, relative paths to the secure server would need to be used to avoid unexpected prompts to be displayed like "Insecure data found. Continue?" Testing will also check to see if the forms that request and collect data are secured, rather than the entire website which can slow down the customer's browsing experience.

Secure Shell (SSH)

Secure shell testing is applicable where SSH is used to log into another computer



over a network. Because it proves strong authentication and secure communication, it is more reliable than telnet, FTP etc.

Since a variety of authentication and encryption ciphers are used by SSH, the exact nature of testing will depend on the combination of authentication and encryption in place.

IPSec

IPSec protocol is popularly used to secure end to end communication, over public and private networks. Because it uses encryption and cryptography, it enables security to be easily customized. Similar to SSH, depending on encryption and algorithms used, testing will aim to uncover holes and weaknesses in the particular implementation in place i.e., modes of operation like transport and tunnel.

Solution

Security testing aims to address inherent flaws and weaknesses in applications to ensure that the final product is robust,

and can safeguard sensitive data without compromising data confidentiality and integrity. Depending on testing time frames, more specific, or thoroughly comprehensive testing can be performed to determine the robustness of an application in the environment within which it will eventually be deployed.

Conclusion

Security testing has gained unparalleled significance over the years, and will continue to rise up the popularity chart given the frequent advancements that we have come to notice and appreciate in the world of technology. Moreover, the importance of data integrity and confidentiality have begun to play a significant role, as have the client demands increased in terms of having access to secure software which delivers without compromising sensitive information. Our aim is to perform thorough security testing to highlight shortcomings and weaknesses in applications to enable developers to build more reliable, and dependable applications in future.