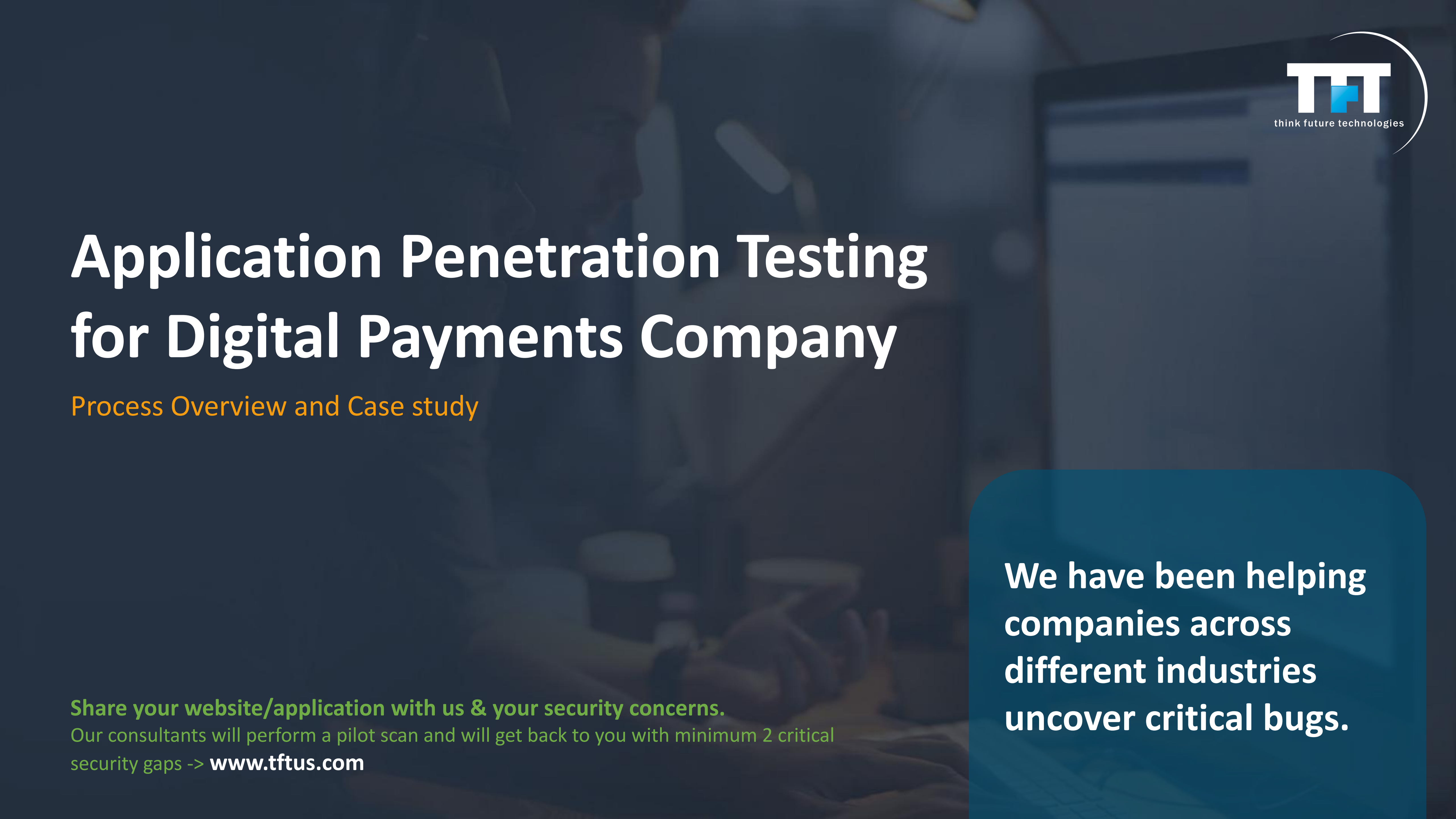

TFT
think future technologies

Application Penetration Testing for Digital Payments Company

Process Overview and Case study

Share your website/application with us & your security concerns.
Our consultants will perform a pilot scan and will get back to you with minimum 2 critical security gaps -> **www.tftus.com**

We have been helping companies across different industries uncover critical bugs.

About

An award-winning digital payments, cash back and analytics company. They help corporates and individuals enjoy gifts and experiences across multiple channels everyday; in the form of employee rewards and recognition, channel partner incentives, employee benefits, online shopping cash back and restaurant cash back.

Challenges

- ✓ Client wanted to launch the product in Web, Play store and Apple Store. As the application carried digital payments. So for it to be considered 'Protected', it needed to pass vigorous tests performed by them

- ✓ Client had an in-house team who have been focusing on the security architecture from the scratch of the product development. In addition to that, client required a security audit to be completed by a third party.

Key Highlights

- ✓ Each bug found was fixed under couple of hours.

- ✓ Software developers(skilled in standard and security development techniques) reviewed the remediated software to assure the quality of the repair and verify no additional security flaws had been introduced.

- ✓ A retest verified the repair of the high risk flaws.

Key Vulnerabilities Found

Found decryption keys through reverse engineering

Includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property

Insecure Data Storage

This covers insecure data storage and unintended data leakage. Failure to identify the user at all when that should be required. Failure to maintain the user's identity when it is required.

Code Tampering

Covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.

Share your website/application with us & your security concerns.
Our consultants will perform a pilot scan and will get back to you with minimum 2 critical security gaps. -> **www.tftus.com**