**Web Application Penetration Testing Methodology**

Web application penetration testing involves simulating cyberattacks against application systems (APIs, front-end servers, back-end servers) to identify exploitable vulnerabilities and access sensitive data. It helps companies verify their systems’ security, identify any vulnerabilities and their scope of the damage, and develop strategies to mitigate potential threats.

The following is a general workflow of an external Pentesting Engagement:

1. **Preparation/Planning**: The first step of the manual application security testing methodology is to determine what information you need; this will depend on how much knowledge you already have about your target. For example, if it’s a new website then you’ll probably want to start by doing some reconnaissance on the business to learn more about their website, who runs it, and how you can contact them. You’ll also need to create a plan for testing by deciding what techniques will be used (black-box, gray-box, or white-box), and which targets should be tested first.

2. **Discovery/Reconnaissance**: The first step in any penetration testing is reconnaissance, where the tester gathers as much information about the target system as possible. This includes identifying open ports on systems, running scans for public data (such as banner grabbing), and looking for any clues that may help in later stages.

3. **Enumeration**: Once reconnaissance has been completed, the next step is enumeration which involves attempting to gain access to resources that were identified during the first stage. This often includes using brute-force methods to try and guess passwords, trying common exploits against known vulnerabilities (such as SQL injection), and dictionary attacks against user names.

4. **Scoping**: This phase includes creating an initial test matrix, defining the scope of each technique used, and deciding what will be tested first, etc. You should also create a list containing all URLs that you’ll need to crawl through, then prioritize them based on how interesting they are likely to be for your testing

5. **Attack Surface Analysis**: This is where you’ll perform a deeper analysis of the URLs discovered during reconnaissance, identifying all possible attack vectors such as common web vulnerabilities (XSS, SQLi, etc.) and OS command injection attacks that could be used to compromise your target. You should also identify any interesting services or directories that may contain sensitive files.

6. **Vulnerability Analysis**: This is where you’ll focus on identifying specific vulnerabilities in the target web application. You should use a combination of manual and automated techniques to do this, such as using a vulnerability scanner or fuzzing tools.

7. **Privilege Escalation Attacks**: Once you’ve found some juicy targets (vulnerabilities) it’s time to start exploiting them! This is where privilege escalation attacks come in handy, as they can be used to gain access to systems or data that would otherwise be unavailable.

8. **Exploitation**: In this phase, the tester actually attempts to take control of the system by using information gathered from previous stages. This may include installing backdoors, creating new accounts with elevated privileges, or simply stealing data from the target system.

9. **Containment**: This phase involves assessing the impact of any vulnerabilities that were discovered and determining how to prevent them from being exploited. The tester must also determine what steps need to be taken in order to restore systems back into their original state if possible.

10. **Cleanup/Recovery**: After exploitation has been completed, it is important to clean up any mess that may have been left behind. This includes removing backdoors and tools that were installed, deleting test files, and restoring systems to their original state.

11. **Reporting**: Once the test has been completed, it is important to document any findings along with recommendations for remediating any vulnerabilities that were discovered as a result of the testing, in a report. You should also include advice on how to protect against future attacks. This can be used by the organization as evidence of its security posture and justification for ongoing spending on security in order to improve it.

Web applications are the primary source of business for numerous companies. With thousands of transactions taking place every second, securing these applications from attacks and data theft becomes crucial. Web application penetration testing can help organizations achieve the highest system security and prepare for any potential threat.