EIONEUS

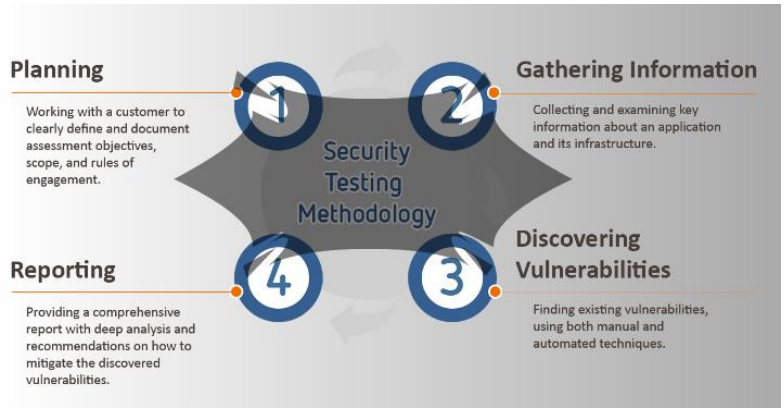
Test Case

This is a case study of an external network penetration test that Eioneus performed on the IT infrastructure of a mid-size organization. Some of the information has been modified to maintain confidentiality.

Overview

The client approached Eioneus to review the security posture of their IT infrastructure from the Black box standpoint. The client shared an IP address range which had servers, network devices and web applications. With no information of the type of systems deployed,Eioneus accepted the challenge and started its assessment.

Any typical security assessment incorporates following workflow which is in line with the Eioneus strategy.



The methodology has been designed exclusively keeping in mind the PTES standards which covers detailed execution plan and strategy for complete coverage and quality.

PENETRATION TESTING EXECUTION STANDARD
PTES
Technical Guidelines

With the help of a tool like Nmap, all active machines were mapped quickly following a Nessus scan on the provided IP range, exposing a bunch of vulnerabilities on network devices and web servers, which further aided in understanding the nature of vulnerabilities and their severity. To minimize the false positive every machine was carefully investigated to confirm the vulnerability existence using manual approach - probing the open ports and gathering Intel on running services. Further Service versions were identified and mapped to vulnerability databases like CVE, NVD; etc. to confirm the vulnerability.

Once the confidence was gained for respective vulnerability, stage 2 unlocks which comprises of exploiting the flaws and gaining shell, privilege escalation, file upload, denial of service; etc. based on the type of vulnerability found on different systems. Exploiting a machine requires fair understanding of the vulnerability and exploit code available. Minor/major changes to exploit code were necessary for successful exploitation.

Various Exploit databases host exploit code for publically disclosed vulnerabilities which can be leveraged for the purpose of exploitation. Tools like Metasploit, Core impact, Kali Linux and several open source platforms and tools came in handy for all penetration testing needs.

Issues Exposed

A majority of security misconfiguration issues like default credentials, open SMTP relay, web configuration files, unauthenticated control panel, tomcat manager console were brought to light. Leveraging these trivial issues granted access to several network resources like routers and web servers which led to complete compromise. Information from Nessus was utilized to exploit remote code execution and buffer overflow vulnerabilities in various systems using Metasploit and publically available exploit code.

Careful examination of the hosted applications exposed SQL injection vulnerability. Exploiting the SQL injection vulnerabilities helped the analysts to obtain some critical business information, in addition to the critical application information.Keeping in mind the structure of the application, security test cases were designed and executed carefully to cover all parameters and important business logic for complete coverage and peace of mind from security threats. Reconnaissance was performed to extract and gather all important Intel's which helped gain deeper understanding of the application and its functions. It didn't take us much time to observe that application is accepting input from user and echoes unmodified in application's response which raised our suspicion for Cross site scripting. There are multiple instances of improper input validation available across the application which can escalate into full-fledged cross site scripting vulnerability.

For example creation of workflows gives user flexibility to provide desired name with no client/server side validation which should eliminate unwanted characters like <>()'/ which are generic characters found in cross site scripting JavaScript attack vectors.

Moving ahead, we came across another interesting find which is common and devastating in application security arena. Cross site request forgery is commonly abused and often deadly. The application had the full potential for CSRF attack but the custom x-session token implemented in the header saved the day. Usually in a request forgery, request is forged/created/crafted (HTML Page) for specific function, let's say for adding a new workflow in automation edge and sent to authenticated user which opens the Crafted html page and process the request and a new workflow is created.

However with the custom addition of x-session token the forged request can't obtain the value for token which results into Unauthorized: invalid or stale authentication token error thus saving the application from cross site request forgery. It is still recommended to implement anti-csrf token as an additional security in case a dedicated hacker finds a way to bypass token mechanism or in a scenario where application deprecates the use of x-session token in header.

Several other issues were highlighted like cross site scripting, session management issues, invalidated redirects; etc.

Remedial guidance was provided for each of the identified weak areas in the tested applications. OWASP top 10 guideline was kept in mind while assessing these applications and designing the mitigation strategy.

Reporting

The executive summary specified that the overall security represented critical risk, and highlighted that although firewall configuration was well maintained, application and operating system security allowed remote intruders to gain access and control to a number of servers.

The number of issues identified at each risk level (critical, high, medium, low and informational) was presented graphically and key issues starting with the most critical were listed with recommendations given for resolution of each.

EIONEUS

Presentation

Delivering detailed report to the organisation, Eioneus interacted with client, making them aware of complete know-hows of report, thereby highlighting every key issue. Following this, the organisation fixed the reported vulnerabilities and then requested Eioneus to carry out a follow-up part of penetration test which involved verification of the fixed issues in the network as well as web applications.