INSIDE OF
ETHICAL HACKING

SWIPE

Unlocking the Secrets of Ethical Hacking and Bug Bounty Programs

Pro CISO®

Contact use here, or visit our website!

+31202117467

1

UNDERSTANDING ETHICAL HACKING

Ethical hacking is all about intentionally breaking into computer systems, networks, and applications - but for a good cause. Unlike malicious hackers, ethical hackers, often called "white hats," get permission to test for security weaknesses. Their main aim is to find and fix these vulnerabilities before malicious hackers can exploit them.

Ethical hackers usually follow a structured process that includes planning, gathering information, scanning for issues, assessing vulnerabilities, and sometimes even exploiting them in a safe, controlled way. Afterward, they provide a detailed report outlining what they found and how to fix it.

2

WHAT IS BUG BOUNTY PROGRAM?

A Bug Bounty Program is an initiative by organizations to encourage security researchers and ethical hackers to find and report security vulnerabilities in their software, systems, or applications. Participants in these programs are often compensated with money, recognition, or other rewards based on the severity and impact of the vulnerabilities they discover. By involving the skills and creativity of a diverse group of ethical hackers, organizations can continuously monitor and improve their security, making bug bounty programs a cost-effective and efficient complement to traditional security measures.

3

ETHICAL HACKING METHODOLOGIES

To ensure a thorough evaluation of an organization's security posture, we have listed common Ethical Hacking methodologies.

Penetration Testing

Planning

- Identification of systems, networks, and applications to be tested.
- Information Gathering using OSINT tools like Shodan and Maltego to collect IP addresses, domain names, and public data.
- Prioritization potential threats based on risk levels.

Scanning

- Network Scanning (live hosts, open ports, and running services)
- Vulnerability Scanning using Qualys or OpenVAS
- Gathering detailed information on network resources, user accounts, and shares.

Gaining Access

- Exploitation of system vulnerabilities, such as buffer overflows or SQL injection.
- Automatic Brute Force to systematically guess passwords, targeting SSH, RDP, or web login interfaces.

Maintaining Access

- Extracting privileged credentials from memory with Mimikatz.
- Verifying the possibility of installing persistent access tools like Netcat for reverse shells or custom rootkits to maintain control.

Analysis and Reporting

- Data Exfiltration
- Documentation of detailed findings, including CVE references, exploited vulnerabilities, payloads used, attack vectors, and remediation steps.

Network Security Testing

Network Mapping

- Topology Discovery
- Identification of running services and versions using tools like Nmap.

Firewall and IDS/IPS Testing

- Testing firewall rules with techniques like packet fragmentation or source port manipulation.
- IDS/IPS Evasion using encrypted payloads or obfuscation techniques to bypass detection systems.

Wireless Network Testing

- WEP/WPA Cracking using Aircrack-ng to crack Wi-Fi encryption keys.
- Setting up rogue access points to intercept and analyze network traffic using tools like Wireshark.

Application Security Testing

Static Application Security Testing (SAST)

- Analyzis source code using tools like SonarQube.
- Automated Scanning using static analysis tools to scan the codebase for security flaws.

Dynamic Application Security Testing (DAST)

- Testing applications during runtime using tools like OWASP ZAP or Burp Suite.
- Identification of input validation flaws.

Interactive Application Security Testing (IAST)

- Combination of SAST and DAST techniques using tools like Contrast Security

Manual Testing

- Manually testing application logic to identify flaws that automated tools might miss.
- Testing custom exploits for identified vulnerabilities to assess their impact and feasibility.

THREAT MODELLING

4

Threat modelling is a proactive approach for identifying functional vulnerabilities by simulating potential attack vectors and understanding the motivations and capabilities of potential adversaries.

Web Application Security:

Conducting threat modelling on a web application to identify and mitigate vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure authentication processes. This helps prevent attacks that could compromise user data, financial transactions, and overall system integrity.

Cloud Infrastructure Protection:

Performing threat modelling on a Cloud infrastructure to detect potential threats such as misconfigured access controls, inconsistent conditional access policies, and exposure of public repositories. This ensures robust defences against data breaches, unauthorized access, and service disruptions in a cloud environment.

Mobile Application API Security:

Analysing a mobile application that accesses an API in the cloud to uncover threats such as insufficient authentication, data leakage through unencrypted API calls, and improper session management. This protects sensitive user information and maintains secure communication between the mobile app and cloud backend services.

We're just getting started, and there's much **more** to explore!

At **Pro CISO®**, our expertise lies in enabling organizations to methodically develop and enhance cybersecurity measures. With our holistic approach to cybersecurity we support companies to develop a secure culture throughout the organization by making use of available resources and integrating with the existing culture. We listen to our customers and tailor security awareness projects to the needs of the company, the people and the relevant threat actors.

Additionally, we provide tactical advice for immediate impact, addressing clear risks while aligning with broader strategic objectives.

Reach out to learn more about how we can help you!