PENTEST, PENTEST, PENTEST

Penetration testing (pentesting) is in essence an ethical hack of a system, also known as an authorized simulated cyberattack, which can be categorized into three types: white box (known information about the system), grey box (system partially known), and black box (system fully unknown). Modern-day pentesting is highly regulated and follows structured frameworks, however its origins go back to the mid-1960s when time sharing took off. By the late 1960s, the U.S. Department of Defense (DoD), National Security Agency (NSA), and Central Intelligence Agency (CIA), as well as academia and industry, had come together to assess and confirm the threat that computer penetration posed. By the 1970s, formal "tiger teams" entered the scene with the sole job of pentesting.

Why pentest?

- **Risk management:** When a vulnerability is found, it will be accompanied with some kind of scoring; there are scoring systems such as the Common Vulnerability Scoring System (CVSS), but ultimately, any score is usually made up of ease, likelihood, and impact. Based on this score, it can then be grouped into a risk category of Very High, High, Medium, etc. This plugs directly into risk programs and can help direct priorities.
- **Security control validation:** There is no better way to validate your current controls than to pentest them, which can determine whether they are effective or not and identify strong and weak points within an enterprise. This can also spark other conversations, such as which solutions are out there to help.

- **Compliance:** Some regulatory standards mandate regular pentesting as part of the effort to be compliant. This shows that due diligence is being demonstrated toward securing their systems.
- **Playbook preparation/incident response:** Simulating an attack also gives you an opportunity to stress test process and early detection systems. If the simulated attack is detected, it can be challenged against what should happen, e.g., go to the security operations center (SOC) and, once at the SOC, whether the ensuing process was effective at responding to an attack.
- **Third-party testing:** The next attack is in development somewhere. No doubt, the internal team is very good, but it is very hard to keep up with the experience a third-party pentest team will gain from all the organizations they test. Validation using external pentest teams gives an unbiased evaluation and demonstrates a commitment to keeping the systems protected.
- **Awareness:** Nothing will get attention quite like the pentesting team breaching the main system of record using a technique that's very easy and likely to exploit. Not all stakeholders are technical, so putting vulnerabilities into a language that is digestible brings everyone onto the same page. Having a vulnerable posture validated also factors into commercial decisions and budget allocated to remediation.
- **Identify vulnerabilities:** Knowing where vulnerabilities are in your system means an organization can address them before they are exploited. Depending on how mature the organization's security journey is, having a list of vulnerabilities can be an effective way to direct remediation efforts.

Mainframe pentesting

We now know why an organization should pentest, so let's talk about the mainframe and pentesting. Unlike many other platforms in the enterprise, there is no specific toolkit to test the mainframemost of it will be homegrown tooling. Ultimately, a pentest provider can tailor a bespoke service with you, but I will talk through which services are available for the mainframe and what three mainframe pentests could look like in the context of white, grey, and black box.

Types of mainframe pentesting:

- Network: internal, and external
- Operating system: IBM® z/OS®
- IBM Enterprise Content Management System Monitor (ESM) Security Controls (IBM® RACF®, Broadcom ACF2, and IBM® TSS)
- Subsystem-specific:
    - IBM® CICS®
    - IBM® MQ®
    - IBM® IMS™
    - IBM® Db2®
    - IBM® WebSphere Application Server® (WAS)
- **White box example:** System access is given with a basic user ID and password for a CICS application. The tester is told how to navigate to the application, and the scope of the pentest is a given CICS application only.
- **Grey box example:** System access is given with a basic user ID and password (or just the IP

address of the mainframe). The tester is given limited knowledge of the systemfor example, whether it runs Db2. The tester can use any techniques to take over the system.
- **Black box example:** A black box test may have several layers, with one team doing a physical breach, a specialized network pentest team or social engineering team getting to the point a mainframe can be accessed, and the mainframe pentest team then taking over. From this point, the tester will attempt to breach the mainframe, from sniffing credentials to brute-forcing, all attack vectors will be tested.

Security assessment versus pentesting

A security assessment is an in-depth collection process of configuration and controls to ultimately report on where improvements can be made. A pentest, however, may daisy-chain together several of these weak points to carry out an attack. An assessment suggests where an organization could be vulnerable, whereas a pentest proves it. An assessment will be comprehensive, while a penetration tester may stop and deem the test a success once they are in. Ultimately, both are important, offer different things, and should be carried out regularly. One blind spot an assessment can present is an instance of three low-risk, misconfigured settings that combine to create a high-risk vector for the penetration tester.

Potential concerns with penetration testing

There are also potential drawbacks to pentesting, and it is important to discuss them.

- **Disruption of operations:** If not carefully planned, or if the system is not properly understood, it can cause disruption (this is unlikely with an experienced team and a well-planned test).
- **Scope limits:** If the client limits the scope very tightly, it can miss vulnerabilities in other areas of the system and give a false sense of security (false positive).
- **Resources:** Specialized resources are very few and far between and finding someone who can test your system can be a challenge.
- **Frequency:** The period during which the tests are conducted needs to be on a sensible cadence, considering how likely new and evolved threats are to enter the landscape.

The future

Automation and artificial intelligence (AI) will inevitably play a role going forward. As penetration testers gather more data, it can be trained into models to suggest new vectors. AI models themselves will be tested; as models have a goal, this goal can be exploited and “poisoned.” Zero Trust models may be pentested, rather than the system itself, to show that an organization is not giving privilege to any user. Organizations will also adopt continual testing as toolkits are developed. It is inevitable that a solution will enter the market that mimics a long-running started task and constantly checks whether an exploit can be executed.

Summary

While penetration testing should be a cornerstone of improving your security posture, as with most things, it is only effective if action is taken afterward. Pentesting alone is not adequate to keep your system security validated; it should be accompanied by assessments, as well as security products such as real-time alerting and compliance. This is a space that will continue to grow on the mainframe. With regulations such as the European Union’s Digital Operational Resilience Act (DORA) going into effect soon, if you aren't already pentesting, get ahead of the regulations that are mandating it and start the process yourself.