Penetration Testing: The white hat hacker

By Vincent Lui

Penetration testing is properly defined as the simulation of an attack against a target network or application, encompassing a wide range of activities and variations.

Most people attempt to define penetration testing as a network attack against an Internet DMZ with the goal of breaking into the internal network. Fundamentally, however, penetration testing is properly defined as the *simulation of an attack* against a target network or application, encompassing a wide range of activities and variations. Some of these variations include simulating an insider threat as opposed to an external attacker, varying the amount of target information provided in advance of the testing, and deciding whether the IT security staff will be made aware of – *and possibly react to* – the testing.

Why penetration testing?

Penetration tests are typically requested to perform final security checks against critical production systems, to validate a security vendor's SLA (e.g., managed intrusion detection services), and of course, to meet compliance requirements such as PCI/DSS. Once the goal is decided, the desired outcome from the assessment is determined. The importance of having a clear and achievable objective before beginning the entire testing process can not be understated. For example, an achievable objective might be to gauge the risk of an external attacker breaching the security of certain key assets. It is equally important *not* to conduct a penetration test for the wrong reasons, such as trying to identify all security issues within a target. By design, penetration testing is meant to simulate a real-world attack, which means only finding *one way* into the target, not every possible way – once in, he is in. This is not to say that a testing team will only try to find a single entry point, but finding every possible exposure should not be expected. Penetration testing is a targeted attack; it is not effective to try and find every issue with this technique.

With objectives determined, the parameters of the testing can be tailored to guide the fieldwork in the right direction. The process then entails three basic phases: pre-assessment, assessment, and post-assessment.

Pre-assessment

Although performing the pre-assessment activities thoroughly will not ensure a successful assessment phase, *not* conducting it will inevitably result in an unsuccessful (and unnecessarily risky) assessment phase. In pre-assessment, the customer and assessment manger work together to determine the scope of the engagement, to collect necessary testing data (such as IP addresses or user credentials), and to understand the potential risks and impacts of the testing. Typical risks include crashing a server or application, modifying production data on backend systems, and inadvertently disclosing sensitive information. All scenarios require customer (e.g., a system administrator) intervention on behalf of the testing team, and the impact can range from a quick server reboot to the full restoration of an active data store. Furthermore, penetration testing is usually performed against production environments, so it is important to have a backup plan in the event of any disruption and to obtain the proper executive approval before engaging in any fieldwork.

While the majority of the work occurs in the next phase, it would be foolhardy to marginalize the importance of the assessment manager who must work closely with the customer to drive requirements and gather the right information while also understanding the potential risks. Most qualified assessment managers come from a penetration testing background where they developed and refined their ability to evaluate risk. In the author's experience, when you trace back the reason behind most penetration testing mistakes, you will more likely find yourself facing an inexperienced assessment manager who gave approval for an unnecessary activity rather than a renegade assessor taking things into his own hands. That being said, a qualified testing team will take the necessary precautions to reduce the chance of a testing disruption. This includes properly configuring automated scans, carefully validating targets before running exploitation tools, and

simply being extremely careful when performing manual testing.

Length of testing times can vary widely, depending on the criticality of the assets being tested and the project budget. Any system can be compromised given enough time. A good rule of thumb is "two people, two weeks" for more critical engagements and just one week for less important assets.

Finally, the manager should also review the scope and the testing parameters with the team before beginning fieldwork. Once this is accomplished the assessment team can begin the fun part – trying to break in!

Assessment

The customer and assessment manger now take a back seat – other than handling project status updates and any escalation that may arise from testing.

It is common practice for testing teams to provide a 24-hour notification window if critical issues are discovered which require immediate remediation. For example, upon discovering a web page containing the company's entire directory and personal information, the team would immediately notify the customer to remove it. This situation is a perfect scenario because removing the exploit does not affect the team's ability to test and continue exploiting. However, if the vulnerability was a missing patch that allowed remote exploitation of a critical server, they could (1) exploit the issue, (2) collect all the data they needed, (3) setup a means of regaining access not through the missing patch, and then (4) notify the customer. This allows the team to continue simulating real world activities and close the critical gap at the same time.

In addition to helping the continuity of the testing, the 24-hour notification window gives the testing team some leeway to determine if a critical issue really is "critical" and take into account any mitigating factors. Generally, however, testing teams notify as quickly as possible. Unless an extremely compelling business case exists otherwise, this should always be required.

The fun begins

The team kicks-off the assessment by reviewing all provided documentation before starting any passive information gathering. This may include querying third party IP registrars for targeting data and the popular Google hacking techniques.[1] Next, the active information gathering process begins by employing techniques known to elicit rich target information such as DNS zone transfers and limited port scanning of IP ranges. At the same time, the testing team will attempt to fingerprint any available platforms and services. This type of scanning and fingerprinting can result in detailed operating system information as well as open services and their specific versions. An example port scan may reveal that WU-FTPD

version 2.6.0 is running on a Linux server or that a web server is running IIS/5.1 over Windows XP Professional. Information gathering, limited port scanning, and fingerprinting activities create a basic attack surface (or map) of the target environment, which can be analyzed by the testing team to identify potential targets. A common technique is to identify already known vulnerabilities in running services such as the SITE EXEC overflow exploit against WU-FTPD 2.6.0,[2] which is available in popular open-source penetration testing tools.[3] Based on the available services, the team may also attempt to identify any system misconfigurations or perform light brute-forcing against services such as FTP or telnet.

If low-impact testing does not yield any viable exploitation paths, the team may elect to conduct noisier information gathering techniques including more comprehensive port scanning, which is usually performed with automated vulnerability scanning tools. This process includes performing the scanning and fingerprinting to create a map of the attack surface and then comparing the map against a database of known vulnerabilities. Based upon configuration, the scanner may also attempt limited brute-forcing or attempt to identify misconfigurations. While automated scanners provide a definite speed and convenience advantage, the downside is that they are prone to false positives and, worse, false negatives. As a result, any automated scan results must be manually verified to ensure their accuracy. On the flip side, automated scanning tools can be used effectively depending on the qualifications of the penetration tester.

Any penetration tester should be able to take automated scan results and validate them – making a best effort to vet each potential issue. More qualified penetration testers possesses a depth and breadth of knowledge that can aid in more accurately validating any identified issues. More knowledge, however, has its limits. What makes the best penetration testers stand head and shoulders above the rest is the talent to synthesize several pieces of disparate information and produce meaningful relationships. From the volumes of information, the best testers can *visualize the paths* through each environment and know what will be required to systematically test, leverage, and chain together the issues to successfully compromise the target. This is not unlike a chess grandmaster who has the ability to examine a board and immediately know the next move. It is in creating these potential exploitation paths where automated tools fail miserably. So while you may be able to replace fair and good assessors with more accurate checks, you will never be able to replace the best assessors.[4]

Once the testing team has validated the automated scan results and developed potential attack vectors, the team will systematically begin evaluating each path, constantly updat-

1   http://johnny.ihackstuff.com/ghdb.php

2   http://osvdb.org/displayvuln.php?osvdb_id=11805

3   http://www.metasploit.com

4   http://www.blackhat.com/html/bh-usa-07/bh-usa-07-speakers.html#Moore

ing their attack strategy with every piece of new information in order to create better paths, close paths, and build variations on existing paths. There are generally three goals should the team successfully compromise a target:

- Ensure continued access to the target
- Access available resources (information and computing power)
- Attempt to leverage the system's trust relationships with other assets

For example, the team may compromise a Windows server by means of a weak system account password and then create an additional administrative account to ensure continued access. Next, they scour the system files for any sensitive documentation and retrieve the password hashes for off-line cracking. Due to the pervasiveness of password reuse, oftentimes an administrator password will be the same on other systems, so access can be gained on these systems as well. Additional techniques may include keystroke logging, network sniffing, and browsing for sensitive information within log files and data stores. If only user-level access is gained, they will attempt local penetration testing within the system to escalate their privileges and gain access to sensitive information. This process of exploitation continues as the testing team systematically progresses through the systems in an attempt to achieve all of the testing objectives. This phase (and the fieldwork) concludes when the testing objectives are reached or the testing team runs out of time.

Post-assessment

Upon completion of all fieldwork, the assessment manager generates a consolidated status update that covers all the activities performed during testing. The testing team as well drafts the initial report for the assessment manager's review. The manger then meets with the customer to go over the issues in the report, to identify any mitigating factors, and to eliminate any false positives. The customer may optionally choose to hold an additional out-brief with a larger audience. Finally, the assessment manager and the customer tie up any loose ends and wrap up the engagement. Objectives met. Test complete.

Summary

Goals and objectives have been set. The penetration team has assessed the systems. The final report identifying security issues has been presented. Used properly, a penetration testing report can help effectively drive change within an organization as well as secure the systems tested.

About the Author

Vincent Liu, CISSP, CCNA, is the Managing Director at Stach & Liu, an IT security consulting firm providing professional services to the Fortune 500 and global financial institutions. He is a contributor to the Metasploit Project, has presented at Black-Hat, ToorCon, and Microsoft BlueHat and has been published in many interviews, journals, and books. He can be reached at vliu@stachliu.com.