



Color Me Clear: Fallacies of Black Box Penetration Testing **By Larry Detar, Vice President , EC-Council Global Services.**

Pick up the newspaper or browse to your favorite on-line news source on any given day. Frequently, you will find an article about a computer network attack. Organizations small and large continue to suffer major security breaches at the hands of Internet attackers. Yet network attacks are nothing new. Vulnerability discovery and exploitation have been happening for years. Yes, as new vulnerabilities were discovered, enquiring minds wanted to know if the exploit code actually worked and crackers tested that code on networks all over the world. At the same time, application and operating system vendors were inundating system administrators with service packs, hotfixes, patches and security bulletins to mitigate or eliminate the new security holes. Most of the time, within days or weeks of a discovery, patches or mitigation strategies were available for the vulnerability. As long as the network administrator applied the patch or strategy, their network should have been secure.

Unfortunately, that didn't always happen or didn't always happen quickly enough. Business process owners wailed that patch impact assessments took time. They had to analyze the effect that the patch or mitigation strategy might have on their own and connected business processes, installed applications, customer service and, of course, the bottom line. Profit margin and customer service sometimes trumped security. Data breaches escalated and the ensuing clamor from violated constituents began to get the attention of elected officials.

Lawmakers began speaking eloquently on the virtues of protecting personally identifiable information and other sensitive electronic data. Some regulated organizations were required to "prove" that their networks or business processes were secure by having an independent third party opine on their network security posture. They were required to have "outside" testing of their networks conducted as part of their annual financial audit in addition to internal testing. Requirements for network security testing began appearing in regulations governing financial, government, and retail sectors.

The National Institute of Standards and Technology (NIST) produced Special Publication 800-42, "Guideline on Network Security Testing", often cited as the prevailing guidance on penetration testing. The 2003 edition of this guideline states: "The purpose of penetration testing is to identify methods of gaining access to a system by using common tools and techniques used by attackers." It briefly discusses "red" and "blue" teaming methodologies, and covert and overt attack strategies: "To simulate an actual external attack, the testers are not provided with any real information about the target environment other than targeted IP address/ranges and they must covertly collect information before the attack." Although not specifically named as such, this last statement has often been used as an operational definition of "black box" penetration testing and the mandate for conducting that test "just like a hacker would do it." This approach seemed to make sense. If we used the same information that was available to an attacker, and tested our networks with the same tools and techniques that were available to an attacker, we should obtain the same results as an attacker. Simple.

Humans are inexplicably drawn to all things dark and mysterious and the art of cracking and network security testing are no exception. The very name "black box test" is filled with intrigue. The entertainment industry was quick to capitalize on the allure of network attacks, producing movies and television shows



depicting crackers (ethical or unethical) as being able to “pwn” (own) a network with a few quick keystrokes despite the best efforts of security personnel and network administrators. Without having any knowledge of the target, these Hollywood whiz-kids logged onto the Internet and were able to easily compromise network after network, despite the best efforts of the target’s network security team. They unerringly found whatever information they needed, and then used that information for good or evil as the script writers saw fit. It’s no wonder that this media-hyped expectation of quick and easy success by crackers seeped into real-life expectations for penetration tests. In the same way that juries expect crime scene investigators to uncover evidence “just like on CSI”, organizations came to expect that conducting a black box penetration test “just like a hacker would do it” helped ensure the security of their network.

S.E. Hinton said it best in her 1971 novel of the same name: “That Was Then, This Is Now.” “Then”, most attacks came from outside the network and were conducted by people who had little if any, inside knowledge of the target, using tools that required a good deal of expertise. “Now”, half or more of the attacks are waged by insiders who already have at least some knowledge of the infrastructure, using point-and-click tools that can be obtained from any hacking or torrent site. Like it or not, “like a hacker would do it” depends entirely on the hacker. It depends on the hacker’s knowledge, level of proficiency and the information on the target network that is available to them. One of the first things we learn as Certified Ethical Hackers is that there are hundreds, if not thousands, of tools available to attack a network. Some require more knowledge of the victim network than others. Different tools expose different vulnerabilities. Which tools does the hacker know about? Which ones do they prefer? Which ones will they use? Which vulnerabilities have they discovered? Which vulnerability will they choose to exploit? The answers are as varied as the hackers themselves are. What information might an inside attacker already possess? We don’t know. The only certainty is that at some point, our network will be attacked. The idea of testing networks “just like a hacker” begins to lose a bit of luster when these facts are considered.

If you’re a crypto geek, then one of your idols is probably Augustine Kerckhoffs. Kerckhoffs was a 19th century writer and cryptographer who developed six basic principals of cryptography. One of these principals has become known as Kerckhoffs’ Law: “A cryptosystem should be secure, even if everything about the system, except the key, is known.” Bruce Schneier, founder and CTO of BT Counterpane was one of the first enlightened individuals to tie this principal to computer networking systems. In a 2002 article for Atlantic Monthly, titled “Homeland Insecurity”, Bruce says: “Kerckhoffs’s principle applies beyond codes and ciphers to security systems in general: every secret creates a potential failure point. Secrecy, in other words, is a prime cause of brittleness—and therefore something likely to make a system prone to catastrophic collapse. Conversely, openness provides ductility.” To put it another way, “Security systems that utterly depend on keeping secrets tend not to work very well.” If secrets create potential failure points, how many points of failure must exist in the black box testing methodology?

According to NIST, penetration testing consists of four phases: planning, discovery, attack and reporting. Let’s look at a few of the possible points of failure in black box penetration testing through Kerckhoffs Law and Schneier’s insight.

Planning Phase

Network security tests are often administratively controlled by “rules of engagement” that serve as formal permission to conduct the testing and specify how the testing will be conducted. Within the rules of



engagement, the tested organization specifies IP addresses/ranges to be tested. In some cases, the ranges provided are not the complete universe of addresses or ranges used by the organization, but are “representative”, “statistical samples”, or only the ranges available from public domain sources such as Regional Internet Registries (RIR). It is not uncommon for organizations to also lease IP addresses or ranges from Internet Service Providers (ISP) or other organizations in addition to the ranges they may own outright. **Point of failure:** Unless the organization chooses to disclose all ranges in use, undiscovered and untested ranges represent an unknown and unquantifiable threat to the organization.

Clients frequently argue that it is a part of the test team’s job to find these “secret” IP addresses and that they should not have to participate in a “full disclosure.” After all, an attacker would have to search for them, right? True enough. If the attacker were an “outsider” with no previous information about the network, they may search for months to ferret out the organization’s IP ranges. Few penetration tests last that long. The cost of conducting a six-month discovery phase would be astronomical and few clients would be willing to pay for it. If the attacker were a former employee or administrator, they may already know about these additional IP ranges and would have to do no searching at all. It is financially impossible to spend as much time as an attacker might in discovering IP addresses and ranges in use and we do not know what information previous employees may have in regard to the organization’s IP scheme. Therefore, we should not be paying test teams to play scavenger hunt but to find and verify vulnerabilities and test the effectiveness of security controls. The hours used in searching for additional IP ranges could be much more effectively used in verifying discovered vulnerabilities. It has been my experience that organizations sometimes insist on practicing “security through obscurity” if they may be having problems with vulnerability management. For these organizations, having the penetration test team play scavenger is an advantage. The fewer IP addresses the team has to work with, the less chance the team has of finding and exploiting vulnerabilities. “It’s all good” until a real attacker discovers the secret.

The tested organization may (and usually does) specify IP addresses/ranges to be excluded from testing as well as dates and times in which testing cannot be conducted. Despite the guidance from NIST to “Test the most important systems first”, an oft heard reason for exclusion is that “the host is too critical to risk a crash caused by a vulnerability scan or vulnerability exploitation.” The point of a penetration test is to verify possible vulnerabilities, but those vulnerabilities must first be discovered. Seldom will a properly configured vulnerability scan against a host at current patch levels on a well maintained network cause a crash. It makes no sense to designate a machine as so critical that it cannot be tested for the existence of vulnerabilities that an attacker will use to compromise that machine. If a system is that critical to the organization, it should have the highest priority for vulnerability scanning and vulnerability verification.

Some common reasons for “blackout”, “no go” or “off limits” testing dates and times are: “we can’t have our business processes slowed down by testing during business hours” and “there are too many remote users who must have access to the system”, and “our work is too important to be slowed or interrupted by testing.” Aside from the obvious that an attack can occur at any time of the day or night, restricting test windows reduces the chances of discovering vulnerabilities that may only occur while business processes are running or that occur because of a heavy traffic load. An actual attacker may monitor the network at various times over an extended period to determine both process related vulnerabilities and traffic-dependent vulnerabilities. Virtually all vulnerability scanners can be configured for a minimum impact on even the most heavily used network. Although scans configured in this manner may take much longer to



complete, the added time is a negligible price to pay when compared with the security advantage gained by testing the system when it may be most vulnerable to failure.

The rules of engagement are also commonly used to reach agreement on specific types of testing allowed or disallowed during the test. Normally, this will either be explicitly specified during the creation of the rules of engagement or one or more paragraphs will be included that specify a procedure for determining which discovered vulnerabilities will be exploited. In many tests that I've been associated with, this results in a number of vulnerabilities being placed "off-limits" to exploitation for the same reasons that hosts or subnets were originally excluded. **Points of failure:** When systems are completely excluded from testing, vulnerabilities on the excluded hosts remain secret. When testing is limited to non-critical days and times, process-related and application-related vulnerabilities remain secret. When vulnerabilities are placed off-limits to exploitation, their exploit potential remains secret. Regardless of the case, if an attacker discovers the secret, the result can be catastrophic.

Discovery Phase

"The discovery phase starts the actual testing" and usually consists of two parts: port scanning, Domain Name System (DNS), whois, web server and Lightweight Directory Access Protocol (LDAP) searches and banner grabbing are often performed externally and port scans, packet capture, and NetBIOS enumeration internally. Part two consists of vulnerability analysis.

It is in the discovery phase that "secrets" really begin to hamper most penetration tests. As we talked about earlier, some organizations insist that the discovery phase be performed covertly. Their rationale comes from two portions of the Guideline: "To simulate an actual external attack, . . . they [the test team] must covertly collect information. . . ." and "This type of testing [Red Teaming] is useful for testing not only network security, but also the IT staff's response to perceived security incidents and their knowledge and implementation of the organization's security policy."

Most organizations have interpreted these statements to mean the discovery phase should be conducted like a child's game of hide and seek. The test team provides the IP addresses/ranges from which it will be conducting attacks to the organization and the organization follows "normal" detection procedures during the testing. As soon as the team's activities are detected, the activity is reported to the test monitor, and, if confirmed, testing ceases. There is no argument that providing the attack source IP addresses is obviously necessary to keep from escalating the incident response procedures outside of the organization.

SP 800-42 provides detailed recommendations on how the attack source IP addresses should be safeguarded and measures to keep the IT staff unaware that testing is being performed. It doesn't take much imagination to suppose that a "heightened awareness" related to intrusion detection and log review occur during the entire testing period, even though specific testing windows are not disclosed to the client. Anything out of the ordinary may get reported, since the rules of engagement are usually not granular enough to specify more than "when testing activity is detected and confirmed." The increased attention paid to network monitoring during the test period may allow the organization to quickly spot activity from the test team, but does little to measure the organization's everyday response capability. Indeed, it may even



provide a false sense of security to the organization: “We caught you. You would never have been able to do anything in our network because we would have caught you.”

More often than not, discovery of the attack team’s activity signals the end of the discovery phase. Organizations are under the impression that the minute activity is discovered, their incident response procedures would have been initiated and would have contained and eliminated the attack threat. Following that supposition to its logical conclusion, they insist that since the attack team was “caught”, no additional discovery take place and the test proceed with the information already obtained. **Points of failure:** The increased awareness associated with a testing cycle gives a false sense of an organization’s actual response to an attack that may occur outside of the testing window. Stopping the discovery phase before all hosts and ranges have been tested allows an attacker to exploit any undiscovered “secret” vulnerabilities on the untested hosts.

Beginning the discovery phase covertly, to “test the IT staff’s response” is good security practice, however stopping discovery upon detection or after a specific number of IP addresses have been identified may mean that entire subnets with vulnerable hosts remain untested. These neglected hosts and subnets may contain any number of exploitable vulnerabilities ripe for discovery by an attacker who, if discovered on one IP address, simply uses another and another and another to continue their reconnaissance until they have exhausted the organization’s IP ranges or they discover a vulnerability they can exploit. A real attacker may use IP addresses from tens or hundreds of different IP ranges and obfuscate which IP addresses are actually being used to attack the network by using decoy IP addresses – an advantage the test team does not enjoy.. Furthermore, the sequence in which subnets were scanned by the attack team may not be the same sequence that an attacker uses. Therefore, the subnet(s) that the penetration test team could not scan because of the rules of engagement may be scanned first by the attacker, allowing them to discover vulnerabilities that the test team was not able to report and of which the organization may not be aware.

Attack Phase

“Executing an attack is at the heart of any penetration test. This is where previously identified potential vulnerabilities are verified by attempting to exploit them. If an attack is successful, the vulnerability is verified and safeguards are identified to mitigate the associated security exposure.” –NIST SP 800-42.

Rules of engagement usually specify that the attack phase is conducted similarly to the discovery phase, so the same hide and seek concerns are repeated in this phase.

Guidance on conducting a penetration test from the NIST guideline also includes the following: “After identifying hosts on the network that can be reached from the outside, they attempt to compromise **ONE** (emphasis added) of the hosts.” **Point of failure:** Untested hosts retain undiscovered and secret vulnerabilities that can be used by an attacker to compromise the network in the attack as well as the discovery phase.

This “capture the flag” mentality allows several advantages to real attackers. The obvious one being that if the test is stopped after successfully exploiting a single vulnerability, the organization is presented with only one possible result of an exploitation. As the Guideline so succinctly states, “This is why penetration



testing is an iterative process that leverages minimal access to gain greater access.” Capturing the flag by exploiting a single vulnerability leaves the network open to other vulnerabilities that were not tested.

Because they were not tested, no evidence was gathered and the organization places a lower priority on mitigation because they were not used to compromise the network. It is safe to say that an average vulnerability sweep will uncover several, if not many potential vulnerabilities. Limiting the attack team to the successful compromise of a single vulnerability leaves other vulnerabilities untested and ready for hacker exploitation. The untested hosts may lead to completely different sets of data files than the successfully compromised host. A real attacker, with different tools, preferences or skillsets may choose a different vulnerability to exploit – one that the test team did not exploit because of their tools, preferences or skillset. It is a far better option to test all discovered vulnerabilities instead of capturing a single flag. It should also be pointed out that vulnerability scanners view vulnerabilities in a vacuum – they consider nothing other than that specific vulnerability in assigning a severity level. Several “Low” vulnerabilities can be just as dangerous as one “High” vulnerability when they exist on the same host.

Reporting Phase

“Generally, at the end of the test an overall testing report is developed to describe the identified vulnerabilities, provide a risk rating, and to give guidance on the mitigation of the discovered weaknesses”
–SP 800-42

Point of failure: Spin Doctoring. While hopefully an attacker will never have access to the report rendered by the test team to the organization (or will they?), the reporting phase is not immune from its own secrets. Obviously, every organization wants a report written to show their efforts in the best possible light, and with good reason! I’ve conducted penetration tests for organizations where senior management’s annual bonuses and performance evaluations were determined in part by the results of my testing. Every discovered vulnerability was closely scrutinized and any and all possible reasons why the vulnerability should be eliminated or lowered in severity were presented. Perhaps the best way to explain “reporting phase secrets” is to share with the reader some of the comments I’ve encountered during the reporting process.

“Your testing represents a point in time. Yes, you found vulnerabilities, but we would have found them in the next scan cycle.”

“We identified these vulnerabilities before. Since we already knew about them, you shouldn’t count them.”

“We caught you. You wouldn’t have been able to compromise anything because we would have caught you anyway.”

“Yeah, this patch was missing on fifty machines, but it was a single patch, so it needs to be counted as a single vulnerability, not fifty.”

Courtesy of Security Systems Resource International Limited (SSR-i)

Shared Copyright of EC COUNCIL & SSR-i