Mistaking Vulnerability Assessments for Penetration Tests Significantly Increases Your Risk

White Paper

Introduction

As more organizations appreciate the growing threats to cybersecurity, they enter the market for cybersecurity products and services. This fast-growing market, expected by Fortune Business Insights to grow to 366 billion by 2028, however, is often plagued by vague definitions, buzzwords, and promises which exceed performance and miss expectation.

Such is the case with offensive security assessment services. While both Vulnerability Scans and Penetration Tests are valuable services, they are manifestly not the same. Yet some service providers offer a "Penetration Test" which is in fact only a "Vulnerability Assessment" or "Vulnerability Scan." Doing so has several potential serious ramifications. **First**, the vulnerability assessment cannot provide an accurate picture of a client's real-world risks. Later detailed within this report, we show some specific examples of these risks. Thus, overreliance on Vulnerability Assessment results provides a false sense of security because many "low" or "moderate" Vulnerability Assessment findings are easily exploitable by real-world attackers as a steppingstone on their way to gain full system compromise. **Second**, relying upon mislabeled "Vulnerability Scans" as Penetration Tests can bring serious legal trouble.[1] **Third**, in our observation, companies that purchase a Vulnerability Scan mislabeled as a Penetration Test are overpaying for services and work products, sometimes dramatically so, for what they are actually receiving.

Definitions: Penetration Test & Vulnerability Assessment

While there are no single definitive definitions of "Penetration Test" and "Vulnerability Assessment," several prime source definitions highlight the differences between the two.

The New York Department of Financial Services' Cybersecurity Regulation; NYDFS.NYCRR.500, requires annual penetration testing. The Regulation defines "Penetration Testing":

> Penetration Testing means a test methodology in which *assessors attempt to circumvent or defeat* the security features of an Information System *by attempting penetration of databases or controls from outside or inside* the Covered Entity's Information Systems.[2]

Key to this definition is the active participation of assessors attempting to "circumvent or defeat" security features, and to attempting "penetration of databases or controls."

[1] Please note that this whitepaper is for general informational purposes only and is not intended to provide legal advice. Consult competent counsel regarding any and all legal questions.

[2] New York Dept. of Fin. Servs. Cybersecurity Regulation ("NYDFS Regulation"), 23 NYCR 500.01(h) (emphasis added).

Similarly, The National Institute of Standards and Technology (NIST) explains penetration testing by, in part, acknowledging its differences from vulnerability scans:

> Penetration testing attempts to duplicate the actions of adversaries in carrying out hostile cyber-attacks against organizations and provides a more in-depth analysis of security-related weaknesses/deficiencies. Organizations can also use the results of vulnerability analyses to support penetration testing activities.[3]

Thus, penetration testing "attempts to duplicate the actions of adversaries in carrying out hostile" actions; it almost goes without saying that "adversaries" that are "carrying out hostile actions" are not just passively scanning for vulnerabilities. Moreover, as NIST points out, "vulnerability analysis" can be used to "support" penetration testing activities, i.e. such scans are a potential predicate step towards penetration testing, but the scans themselves aren't penetration tests.

NIST explains vulnerability scanning as well:

> Vulnerability scanning includes, for example: (i) scanning for patch levels; (ii) scanning for functions, ports, protocols, and services that should not be accessible to users or devices; and (iii) scanning for improperly configured or incorrectly operating information flow control mechanisms.[4]

Notably, vulnerability scanning does not attempt the actual exploitation of located vulnerabilities, nor does it attempt to analyze whether certain combinations of vulnerabilities present additional exploitation opportunities (i.e. the total risk is greater than the risk from the parts).

The Payment Card Industry Data Security Standard (PCI-DSS) also distinguish between "penetration testing" and vulnerability scanning by defining penetration tests as an attempt to exploit identified vulnerabilities:

> Penetration tests attempt to identify ways to exploit vulnerabilities to circumvent or defeat the security features of system components. Penetration testing includes network and application testing as well as controls and processes around the networks and applications, and occurs from both outside the environment (external testing) and from inside the environment.[5]

[3] NIST 800-53 (Rev. 4), CA-8, Supplemental Guidance.

[4] NIST Special Publication 800-53 (Rev. 4), RA-5, Supplemental Guidance.

[5] Payment Card Industry (PCI) Data Security Standard (DSS) and Payment Application Data Security Standard (PA-DSS) Glossary of Terms, Abbreviations, and Acronyms, Ver. 3.2 (April, 2016).

Mistaking a Vulnerability Scan for a Penetration Test Can Be Hazardous

Vulnerability scanning and penetration testing each have their place in a well-constructed security regime. But, problems arise when vulnerability scanning is mistaken for, or misrepresented as, penetration testing.

Operational Hazards

From an operational standpoint, the primary objective of a penetration test is to identify and measure risk. Vulnerability scans typically do a poor job of doing this, despite the fact that they can be an effective tool for quickly creating lists of potential security issues.

Vulnerability scans typically do a poor job of measuring the true risk introduced by specific security issues. A good example of this is how scans fail to express the risk of enabling Windows' Link-Local Multicast Name Resolution (LLMNR) and NetBIOS (NBT-NS) networking protocols. On the surface these alternative methods for host identification are simply replacements or useful augmentations for Domain Name Services (DNS). However, a malicious actor can easily exploit a flaw in these protocols to trick Windows systems into authenticating with the attacker, allowing the attacker to steal domain credentials. The mere presence of these protocols on the network constitutes a critical threat, since the only way to mitigate the flaws in these protocols is to disable the protocol entirely. These LLMNR / NBT-NS poisoning attacks can easily be executed by unsophisticated attackers using publicly available tools and have a devasting impact in the event of a perimeter breach.

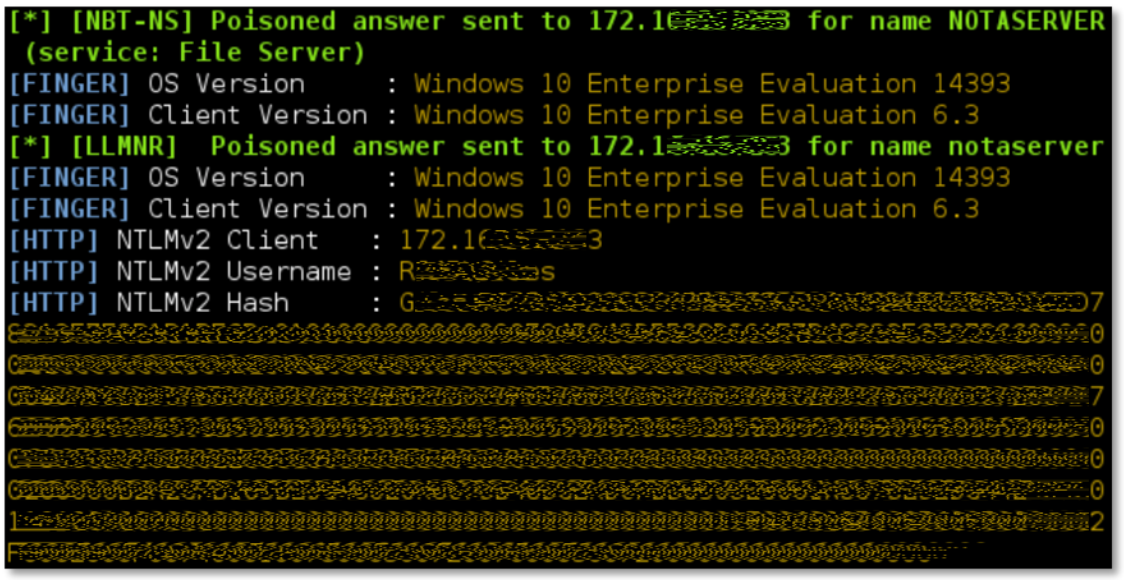
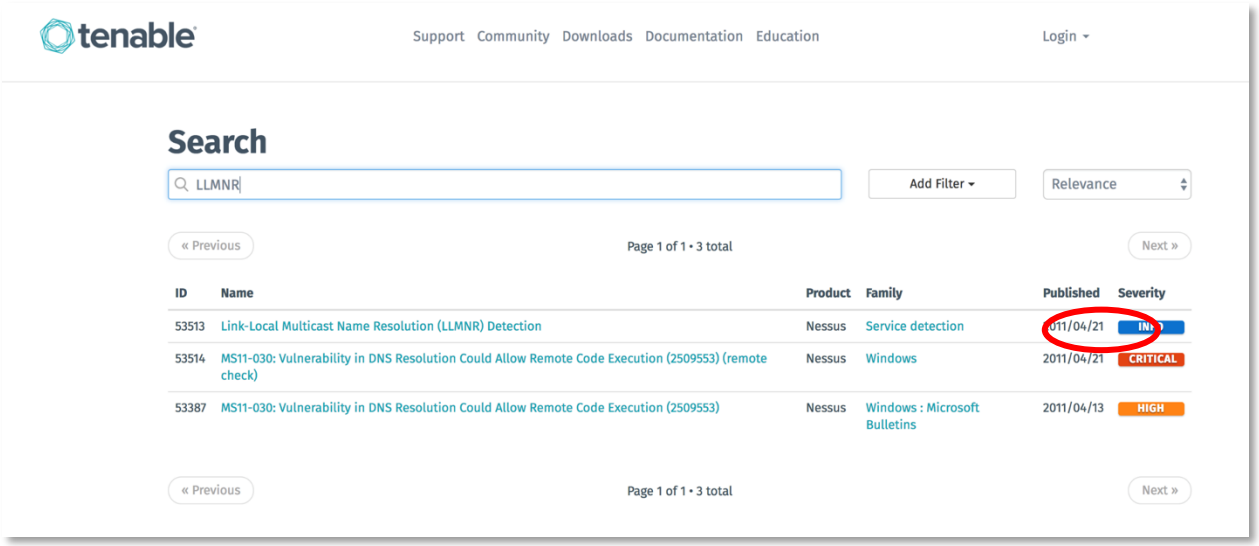
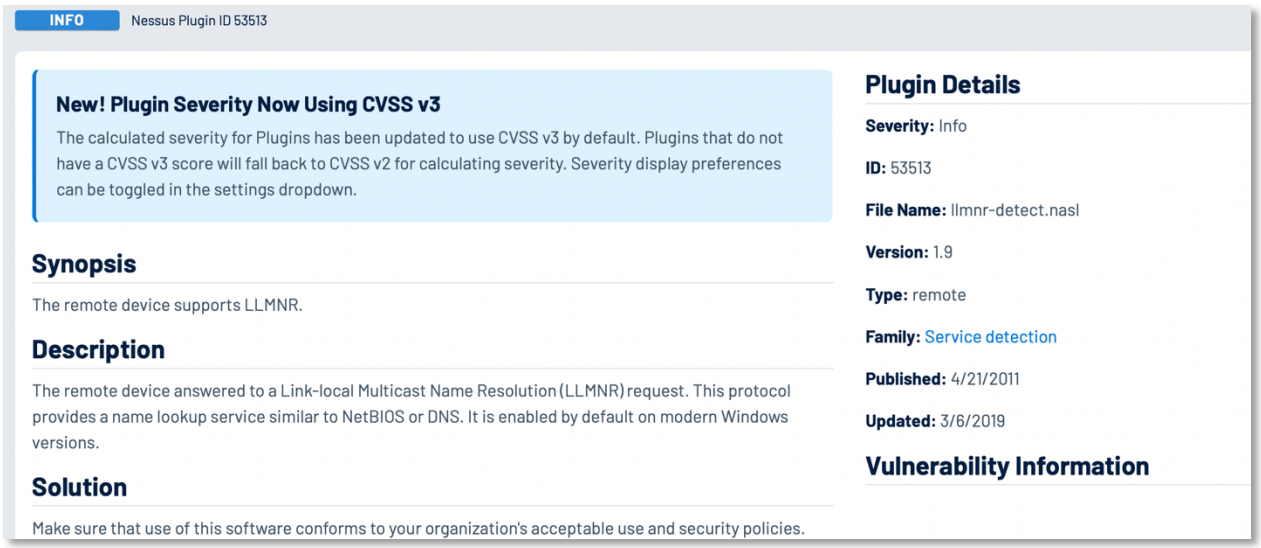


Figure 1 Example Exploitation Script

Recognizing the high impact of such an attack, along with ease at which it can be exploited using publicly available tools, a penetration tester would immediately categorize the presence of LLMNR or NBT-NS as a critical security risk. Yet most vulnerability scanners either significantly under-classify this issue, or not report it at all. Using Tenable Nessus vulnerability scanner as an example, we see that the scanner reports the presence of LLMNR as a “Informational” finding. This means that the finding is not even classified as a risk to the network. Additionally, Nessus does not check for the presence of NBT-NS, let alone assign it an appropriate risk rating.





This is not to say that Nessus is "wrong." Rather, this example simply demonstrates the very real differences between the information provided by a vulnerability scan and a penetration test.

In addition, vulnerability scans cannot effectively gauge the risk posed by a collection of security issues combined with one another. Networks and applications are ecosystems – they are the sum of many moving parts that often rely on each other in subtle ways. Not surprisingly, low severity vulnerabilities that affect related components can often result in a higher risk issue when combined with one another.

To illustrate, take the case of two web application level vulnerabilities that are typically classified as Low or Medium severity findings – Cross-Site-Request-Forgery (CSRF) affecting a login page, and stored Cross-Site-Scripting (XSS) that can only be executed against the attacker's own account. Classifying these vulnerabilities as lower risk makes sense when these vulnerabilities are found in isolation, since the impact of these attacks amounts to self-inflicted inconvenience to the attacker. However, good penetration testers know that finding both vulnerabilities in the same application reveals a High to Critical risk issue. The CSRF vulnerability can be used to force a user to authenticate with the application as the attacker, allowing the self-XSS vulnerability to be executed against other users. This type of combined attack is rarely identified during a vulnerability scan, mainly because of the difficulty involved with automating the combined attack path.

Vulnerability scans will also typically miss more subtle security issues that can affect enterprise applications and networks; for example, insecure trust relationships between Active Directory domains. Consider a scenario involving an investment firm and one of its third-party vendors. Both the vendor and the investment firm use Active Directory to manage their networks. The investment firm has asked the vendor to establish a one-way trust between their domains. This trust would allow the vendor to use its own Active Directory accounts to authenticate with the firm's domain, making it easy to provide the firm with 24/7 product support services.

At first glance, the one-way trust relationship appears to be innocuous from the perspective of the vendor. However, depending on how the firm's domain is configured, this may result in the vendor's account hashes and even plaintext credentials being cached on systems within the firm's domain. This means that the one-way trust becomes what is known in economics as an externality – the vendor now depends on the firm to keep at least some of its accounts secure. An experienced penetration tester will likely spot and report this issue.

Finally, a written penetration test report will not only identify complete attack paths into the organization but will break these paths down into the discrete steps of the kill chain. Each of these discrete steps will be mapped to one or more security issues along with mitigating controls that could have been used to stop that step of the attack, thus breaking the kill chain. By definition, a vulnerability scan cannot be used to provide this kind of information.

Stage	Attack Stage	Result	Contributing Factors	Mitigating Controls
1	External Recon	ACHIEVED ✓	Wireless reconnaissance from considerable distance possible due to AP range.	Reduce access point range.
2	Initial Access	ACHIEVED ✓	Perimeter breached by stealing RADIUS credentials using an evil twin attack. This was possible due to weaknesses in the EAP-PEAP authentication scheme used to protect the network.	Prevent rogue AP attacks using certificate-based authentication (EAP-TLS), Wireless Intrusion Detection System (WIDS)
3	Payload Delivery	ACHIEVED ✓	Performed second evil twin attack to force connection from authorized workstation. Performed SMB Relay from NAC to authorized workstation, granting command execution with SYSTEM privileges.	Require SMB Signing, prevent rogue AP attacks using certificate-based authentication (EAP-TLS), Wireless Intrusion Detection System (WIDS), better endpoint Protection on workstations
4	Payload Execution	ACHIEVED ✓	Powershell-based implant executed on authorized workstation.	Monitor for Powershell usage.
5	Situational Awareness – Local	ACHIEVED ✓		N/A
6	Persistence – Local	NO ATTEMPT ⊘	*Not in scope*	*Not in scope*
7	Establish Command & Control (C2) Channel	ACHIEVED ✓	Reverse SSH shell over port 443 (see: egress filtering).	Filter and monitor outgoing connections.

Legal Hazards

Legal requirements to maintain at least “reasonable” cybersecurity continue to multiply. Starting in 2018 granular prescriptive requirements, such as the NYDFS Regulation’s penetration testing requirement, appeared focused on Financial Services organizations operating in the State of New York. In Q4 of 2021, the Federal Trade Commission released an amendment to the Gramm-Leach-Bliley Act (GLBA) and its Safeguards Rule which is designed to protect consumer non-public information. Under the GLBA’s new Safeguards Rule, organizations with more than 5000 customers will be subject to the same prescriptive penetration testing requirements that Financial Services companies licensed to operate in New York. To the extent these regulatory requirements mandate “penetration testing,” companies who attempt to rely upon vulnerability assessments to satisfy such requirements will likely be found deficient, especially in the aftermath examination after a cybersecurity incident.

Legal risk, however, is not limited to compliance with explicit regulations. More and more business relationships include pre-contractual due diligence and more explicit cybersecurity contract provisions. To the extent that representations or promises are made to contracting counterparties regarding penetration testing (such as statements that a company has undergone “penetration testing”), the

company who mistakes a vulnerability assessment for a penetration test has likely run afoul of those representations or promises. Given the significant monetary impacts of a cybersecurity incident, these types of breaches can afford plaintiff's lawyers significant opportunity to seek large monetary awards.

Entire industries are now also requiring specific testing as part of participation in that ecosystem. Like a dramatic story right out of the movies, Hollywood and the Motion Picture Association (MPA) have the Trusted Partner Network (TPN) program. Under the TPN in order for a vendor to participate with a member studio they are subject to explicit physical, cyber, and content security provisions. These provisions specifically define both penetration testing and vulnerability scanning, and the associated frequency in which a vendor must be assessed.

Even in the absence of contractual representations or provisions, vulnerability scans masquerading as penetration tests increase legal risk. To the extent that a company has a duty to exercise "reasonable" security, the courtroom test for "reasonable" security will be defined by competing expert witnesses. To the extent an expert witness for the opposition can fault a party for mislabeled penetration testing, the overall reasonableness of that party's security efforts looks more suspect in the eyes of the jury:

> Plaintiff's Lawyer: Ms. Expert, can you tell us why you believe the defendant failed to exercise "reasonable security."
>
> Ms. Expert: Certainly. For starters, the defendant lied to its own board of directors, claiming that it had undergone expert penetration testing, when in fact it had only procured a vulnerability scan.

Additionally, we have seen cases where a company had procured cyber risk insurance and after a breach the insurance refused to pay because the company represented that it had undertaken certain security measures but had not done so. Saying that the company conducted regular penetration tests when in fact the only conducted vulnerability scans would be an example of not executing the security measures that were stated on the insurance application.

Wasted Security Dollars

Some vendors charge $10,000 or more for a vulnerability scan labeled as a "penetration test." There are even some companies offering "automated penetration tests".

Project fees for penetration tests vary from firm to firm, and even from region to region. However, for the most part they are dictated by two factors:

1. The amount of time needed to complete the project (computed in terms of person-hours)
2. How difficult it is to find a consultant who is qualified to work on the project (niche technologies, niche industries, highly sensitive testing environments, etc.)

Vulnerability scans are considerably less time consuming than penetration tests, particularly since they are a largely automated process. They also require considerably less skill and experience to execute. With this in mind, charging upwards of $10,000 for a vulnerability scan generally makes no sense, and represents little more than dramatically overpaying for a basic service.

Conclusion

Maturing security programs can make good use of both vulnerability assessments and penetration tests. Each has its place. But confusing the two leads, at best, to wasted security dollars spent on overpriced and mislabeled vulnerability testing. More likely, failing to utilize real penetration testing leads to a false sense of security, and potentially serious operational and legal consequences.