**Vulnerability Scanning with Shields Up and Nessus by Lana Bracken**

**1. What Did You Do?**

To begin my vulnerability scan, I used the Shields Up tool by Gibson Research Corporation (GRC), accessible at <https://www.grc.com>. This tool is designed to remotely probe a system’s public IP address from outside the local network, simulating what a potential attacker might see. I conducted the scans from my home network on my personal laptop, which was directly connected to my ISP-provided router.

From the GRC home page, I navigated to Services → Shields Up, then clicked "Proceed" to begin the scan. Shields Up clearly states that by clicking "Proceed," the user authorizes remote security testing, and it provides assurance that the scans are benign and not stored or misused.

Once on the scanning page, I ran the following two tests in sequence:

1. **Common Ports Scan** - scans frequently targeted TCP ports such as 21 (FTP), 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS), 135/139/445 (Windows services), etc.
2. **All Service Ports Scan** - performs a comprehensive scan of the first 1,056 well-known TCP ports (0–1055).

I saved the output from both scans in plain text format for later analysis.

A few days later, I repeated the scans under slightly different conditions: I had rebooted my laptop and returned from a different network. My public IP address (74.192.65.236) remained the same, allowing for a consistent basis for comparison. I compared the results of the two runs to determine whether my external port visibility changed and why.

In preparation for analyzing the results, I researched the purpose of each scanned port and the meaning of each possible result. Shields Up classifies port responses as:

* **Stealth** – the system does not respond at all (ideal).
* **Closed** – the system acknowledges the port but refuses the connection.
* **Open** – the port is actively accepting connections (potential vulnerability).

My goal was to understand which services, if any, were exposed to the public internet, and whether any unexpected port behavior could indicate misconfiguration or a security risk.

To complete the second part of the assignment, I began by visiting <https://www.tenable.com/products/nessus/nessus-essentials> and downloaded Tenable Nessus Essentials, a widely used vulnerability assessment platform that automates the identification of security issues such as known vulnerabilities, insecure configurations, exposed services, and missing patches. I provided my academic email address to request a free trial activation code, which was delivered via email shortly after.

After downloading the installation package for my operating system, I installed the software. Upon launch, Nessus opened in my browser at <http://localhost:8834/WelcomeToNessus-Install/welcome>, where I entered the activation code, created a local username and password, then the plugin download and compilation process began.

Once the plugins finished compiling, I navigated to the Scans section, selected Basic Network Scan, gave the scan a descriptive name, and entered the target IP addresses. I then saved the scan configuration and launched it by clicking the play button.

This test scan was run on my home network to familiarize myself with the interface, scanning workflow, and report outputs. However, this trial scan used several of the 16 IP slots allowed under the free license, which later impacted my ability to scan the full business network on the first attempt.

With explicit permission from a local small business, the objective was to conduct a vulnerability assessment of their network infrastructure using this tool.

**First Attempt: Business Hours Scan and License Limit**

Next, I visited the small business during regular business hours and connected to their Wi-Fi network to perform the first official scan. The vulnerability scan ran for approximately 30 minutes and completed, but I noticed a message stating that the 16-host limit had been reached. Despite reconfiguring the scan, Nessus had retained the IPs from my home scan, leaving only four usable slots, and thus producing an incomplete picture of the network’s vulnerabilities.

**Reset Attempts and Subnet Configuration Errors**

To resolve this, I returned home to uninstall Nessus, register a new activation code under a different email, and reinstalled the software. I returned to the business after hours and attempted a scan again — this time entering the entire subnet (192.168.20.0/24) into the target field. However, Nessus interpreted this as a command to scan all 256 IPs in the subnet. As a result, the scan once again exceeded the host limit, reporting findings for only the first 16 IPs encountered.

**Third Attempt: Manual IP Entry With Line Breaks**

After repeating the uninstall, reboot, reinstall cycle and obtaining another activation code, I performed an Nmap ping scan to identify 12 active hosts. I then attempted to scan just those IPs by pasting them one per line into the Nessus target field. Unfortunately, Nessus again interpreted this as a full subnet scan, resulting in another exceeded host limit and another incomplete report.

**Final Attempt: Successful Targeted Scan**

Another reinstall and activation process was completed, and I returned to the business once more, again scanning after hours from the parking lot. This time, I entered the 12 Nmap-verified IP addresses into the target field, but separated them with commas, matching Nessus’s required input format. The scan completed successfully, remained within the license constraints, and generated a complete vulnerability report for all 12 intended hosts — fulfilling the assignment requirements.

**Vulnerability Scan Details**

The Basic Network Scan in Nessus used Nessus’s vulnerability database and plugin engine to:

* Identify open ports and running services
* Detect known vulnerabilities (CVEs) based on software versions
* Flag insecure or misconfigured services
* Report unsupported operating systems or outdated firmware

The scan was conducted over Wi-Fi from inside the network, allowing full access to internal devices without interference from firewalls or NAT. After the scan completed, I explored the results through Nessus’s web interface. I examined key fields provided for each finding, including:

* **Severity (Sev)** - Indicates whether the vulnerability is Critical, High, Medium, Low, or Informational.
* **CVSS** - The Common Vulnerability Scoring System numerical score indicating technical severity.
* **VPR (Vulnerability Priority Rating)** - A dynamic score based on exploitability, threat activity, and age of the vulnerability.
* **EPSS (Exploit Prediction Scoring System)** - Predicts how likely a vulnerability is to be exploited in the wild.
* **Name, Family, and Count** - Identify the type of vulnerability, the plugin family it falls under, and how many times it was detected.

Throughout the process, I researched each Nessus metric and plugin family to better understand how vulnerability data is categorized and how risk-based prioritization can optimize how vulnerabilities are addressed.

**2. What are the results?**

I performed two sets of port scans using Shields Up: one targeting Common Ports (frequently targeted TCP ports) and one targeting All Service Ports (the first 1,056 well-known TCP ports, 0–1055). The scans were conducted several days apart from the same public IP address.

**Table: ShieldsUp First Scan Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| Scan Type | Open Ports | Closed Ports | Stealth Ports |
| Common Ports | 0 | 1 (Port 0) | 25 |
| All Service Ports | 0 | 1 (Port 0) | 1,055 |

In the second scan, all ports were reported as stealth. Both scans were conducted from the same IP address, so this change likely reflects a temporary configuration or filtering difference.

The presence of port 0 marked as closed in the first scan is unusual. Port 0 is officially reserved and not used for standard communication. According to research from IEEE and Gibson Research, traffic to/from port 0 is rare and usually indicates raw socket manipulation, fingerprinting attempts, or poorly configured software.¹ The change to full stealth suggests that my firewall or ISP filtered out such traffic during the second scan — a positive sign of improved blocking.

This change suggests a strengthening of firewall behavior or temporary configuration differences between scans. The closed port in the first scan (Port 0) no longer appeared in the second scan, indicating it was dropped entirely (stealthed). This is considered more secure because stealth mode prevents the host from being detected at all via unsolicited probes.

Having all ports in stealth mode makes the network essentially invisible to external attackers. It’s the ideal state for a network perimeter firewall - confirming that unsolicited traffic is silently dropped rather than acknowledged.

**Analysis of Changes Over Time**

The change between scans — from “1 closed port” to “all stealth” — is likely due to adaptive firewall behavior or temporary conditions in the network. For example:

* My router may have updated its firmware or firewall rules.
* My ISP may have adjusted how certain traffic is filtered.
* A device on my network may have briefly interacted with port 0 before becoming inactive.

Shields Up even warns that some firewalls adapt dynamically when they detect scanning behavior, blocking certain probes after the first attempt. Regardless of the cause, the final result of all ports in stealth mode is ideal and demonstrates a strong security posture.

**Nessus Scan Results**

A total of 12 hosts were scanned on the network during the Nessus scan, excluding the scanning device (my laptop), resulting in analysis of 11 unique hosts. Of these, 7 hosts were found to have at least one vulnerability, while 4 hosts showed no detectable vulnerabilities during the scan.

The vulnerability scan revealed several security issues across the network, categorized by severity based on their risk to confidentiality, integrity, and availability. The most severe vulnerability identified was open DNS recursion on host 192.168.20.10. This configuration allows unauthorized recursive queries, which can be exploited to launch amplified Distributed Denial of Service (DDoS) attacks. Immediate action should be taken to restrict recursion to trusted internal hosts.

Other medium-risk vulnerabilities include:

* **Insecure SSL configurations** (RC4 cipher support and self-signed/untrusted certificates) on hosts 192.168.20.89, 192.168.20.1, and 192.168.20.194. These weaken encrypted communications and may expose the network to man-in-the-middle (MitM) attacks. Recommended mitigation involves disabling deprecated ciphers and replacing certificates with those signed by a trusted Certificate Authority.
* **SMB service detection** on 192.168.20.1 via port 445. SMB is a common vector for malware propagation and should be hardened or disabled if not required.

Low-risk findings included:

* **ICMP timestamp responses** from multiple hosts (192.168.20.1, 192.168.20.89), which can leak system time and uptime information. Although minor, this can aid attackers in fingerprinting devices and should be filtered at the firewall level.

To prioritize these findings, I reviewed each vulnerability's CVSS, VPR, and EPSS scores as presented by Nessus. High CVSS scores highlighted technical severity - for example, the DNS recursion issue received a CVSS of 7.5, indicating a serious risk to availability. The VPR added context by weighing real-world exploitability and threat intelligence, helping to differentiate between vulnerabilities that are merely theoretical and those actively targeted. EPSS further supported prioritization by estimating the likelihood of exploitation in the wild; findings with EPSS scores above 0.7 were flagged for urgent review. Together, these metrics enabled a risk-based approach to interpreting the results and identifying which issues warranted immediate remediation.

**Table: Selected Vulnerabilities with Risk Metrics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vulnerability | Host IP | CVSS | VPR | EPSS | Risk Summary |
| Open DNS Recursion | 192.168.20.10 | 7.5 | 7.4 | 0.82 | High-severity issue; exploitable for DDoS amplification. |
| Self-Signed SSL Certificate | 192.168.20.1 192.168.20.194 | 5.9 | 6.5 | 0.62 | Undermines encryption trust; certificates should be replaced. |
| SMB Service Detection | 192.168.20.1 | 4.3 | 5.8 | 0.76 | Enables lateral movement; harden or disable if unnecessary. |
| ICMP Timestamp Response | 192.168.20.1 192.168.20.89 | 2.6 | 2.3 | 0.03 | Minor issue; may aid reconnaissance; filter ICMP at firewall level. |

These results provide actionable insight for network hardening. By prioritizing high and medium-severity findings, the organization can reduce its attack surface significantly. Fixing DNS recursion eliminates a major DDoS amplification vector. Upgrading SSL configurations ensures secure communication, essential for protecting sensitive data. Addressing SMB exposure prevents lateral movement in case of compromise. Regular scans, patch management, and segmentation are tactics that will maintain a proactive security posture and reduce future vulnerabilities.

Although my laptop was included in the Nessus scan for personal awareness, it is not part of the small company’s network and should not be listed among the organization’s vulnerabilities. During the scan, Nessus identified several open ports on my system, including 1433 (Microsoft SQL Server), 3306 (MySQL), and 16992 (Intel AMT). This initially appeared to conflict with the ShieldsUp scan, which showed all ports as "stealth" except for port 0, which was closed. The difference lies in each tool's scanning scope: ShieldsUp scans from the outside, targeting the router’s public IP to check for internet-facing vulnerabilities, while Nessus scans internally, revealing services that are open within the local network. This distinction is important - even if a router blocks access from the internet, services exposed on internal devices may still present a risk if malware spreads through the LAN or if the device is connected to unsecured networks.

**Table: Personal Laptop Vulnerabilities (Not Part of Small Business Network)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vulnerability | Port | CVSS | VPR | EPSS | Risk Summary |
| Microsoft SQL Server Detected | 1433 | 7.4 | 6.9 | 0.78 | Common target for brute force and privilege escalation; should be restricted. |
| MySQL Server Detected | 3306 | 7.2 | 6.7 | 0.74 | May expose sensitive data if misconfigured; consider disabling or securing. |
| Intel AMT (Management Engine) | 16992 | 7.5 | 7.1 | 0.85 | Often exploited in firmware-level attacks; disable if unused. |

**3. What Did You Learn? (Key Takeaways)**

One of the key lessons I learned from using Shields Up is that vulnerability scans provide only a snapshot in time, and results may vary based on a number of factors, including firewall behavior, ISP-level filtering, and temporary changes in device or network configurations. This was evident when port 0 appeared closed in the first scan but was completely stealthed in the second. While researching this anomaly, I came across *The Curious Case of Port 0*, which I found both fascinating and accessible. I especially liked the quote from the article: *"Only because it shouldn’t happen does not mean it does not do so in practice."* It captured the unpredictability of network behavior in a way that stuck with me. The article helped me better understand the unusual behavior of port 0 and appreciate the complexities behind port-level communication. This experience reinforced the importance of performing scans periodically rather than assuming that a one-time check ensures long-term security.

I also learned more about how NAT routers act as a first line of defense for residential users, often masking internal network services from the outside world. However, relying solely on a NAT firewall can lead to a false sense of security. Shields Up provides the option to bypass this protection for testing purposes by temporarily removing the NAT device or placing a computer in the router’s DMZ. While I chose not to pursue that method, since I was using a work laptop and did not want to introduce risk, the recommendation highlights the value of testing individual host security, not just edge device behavior.

Finally, I gained insight into the importance of port configuration and visibility. The goal is not just to block unsolicited traffic, but to do so silently, placing ports into "stealth" mode. This makes the system effectively invisible to external attackers. It's not enough for ports to be merely closed, they should be unresponsive when not in use.

Overall, Shields Up helped me appreciate the layered nature of perimeter defense and the importance of testing not just what is open, but what is visible.

As for this first experience with Nessus, I have to admit, all the extra time trying to get a successful scan has tainted my view of Nessus. I believe a better experience would have been had with a paid subscription. I was humbly reminded that patience and perseverance are essential when working with sophisticated tools like Nessus. Despite their capabilities, tools like these can introduce unexpected hurdles - from licensing limitations to strict input formatting - and overcoming them requires persistence. As someone new to cybersecurity, these challenges were frustrating at times, but they pushed me to learn more and slowly start building confidence with the scanning process.

Another valuable takeaway from this assignment was discovering how helpful the XML export feature in Nessus can be when sharing results with clients. The hyperlinks, structured layout, and color coding make the report easy to navigate, especially for non-technical audiences. I also saw how powerful Nessus can be when targeting specific vulnerabilities, if you know which plugin to use, you can quickly exhaust all issues related to that item. I can see how this makes Nessus a highly efficient tool in larger organizations with complex environments. Compared to Wireshark, I found Nessus’ output easier to read; however, I personally enjoyed working with Wireshark more. It gave me a stronger sense of control over the data, possibly because Nessus’ scanning failures early on made the process feel more frustrating. Despite that, I now see how both tools serve different but complementary roles in network security analysis.

Through this vulnerability scan exercise, I learned that my personal laptop has several ports open internally (e.g., 1433 for SQL Server, 3306 for MySQL, and 16992 for Intel AMT), which are not visible externally through tools like ShieldsUp. This demonstrated the critical difference between internal scans (e.g., Nessus) and external scans (e.g., ShieldsUp). ShieldsUp showed all ports as "stealth" from the internet, while Nessus revealed services exposed within my local network, a critical topic for layered security.

I also learned that although I am reasonably protected on my home network, where the router acts as a firewall, these services could pose a risk if I were to connect to a public network. While I rarely do so - mainly because I rely on large monitors and don’t work well using just a laptop screen - I recognize the increased risk when devices are exposed outside a secured environment.

When I travel, such as staying in a hotel or waiting at an airport, I bring an extra monitor in my suitcase. I also always use my mobile phone’s hotspot instead of connecting to public Wi-Fi. I believe this is a safer practice, as it limits exposure to unknown or potentially compromised networks. This precaution, combined with my awareness of open internal services, helps reduce the attack surface even when working away from my secured home network.

One of the most important lessons I learned from this assignment is the value of understanding your audience when communicating technical findings. In this case, my audience is a non-technical business owner, so it’s essential to translate cybersecurity results into clear, relatable terms. I realized that simply reporting vulnerabilities isn’t enough - what matters is explaining what was done, what was found, what it means for the business, and what needs to be fixed in language that makes sense to them. Instead of using jargon like “open ports” or “cipher suites,” I learned to use analogies and focus on business impact, like comparing a DNS flaw to an office phone being misused or likening expired SSL certificates to fake IDs. This approach helps ensure the business owner understands the risks without feeling overwhelmed or alienated, and it increases the chances that critical security issues will actually be addressed. Technical accuracy must be paired with effective communication; thus technical findings are only actionable when communicated with clarity.

¹ *“The Curious Case of Port 0,” IEEE Xplore.* [*https://ieeexplore.ieee.org/document/8816853*](https://ieeexplore.ieee.org/document/8816853)