Windows System Vulnerabilities

CYSE 280 – Windows System Management and Security Research Paper

Dr. Malik A. Gladden

Old Dominion University

December 01, 2025

Introduction

Windows is one of the most used operating systems in the world, which ultimately makes it a constant target for cyberattacks. Due to the popularity and usage of the software, cybercriminals have utilized different exploits and vulnerabilities to gain advantages financially and politically. Microsoft, the creator and owner of this software and its servers, has pushed out updates and improved security to manage and patch these vulnerabilities to a certain extent, but there is always a constant threat of cyberattacks taking advantage of these vulnerabilities. The Windows system continues to be a challenge as Microsoft has invested huge amounts of resources to improve different aspects like security, software, etc., but cybercriminals continue to target the software as new vulnerabilities and exploits, such as bad patching and configuration, human error, etc., continue to expose the software to be taken advantage of. This research paper will explore how commonly found vulnerabilities and exploits are taken advantage of by referencing historical and past examples of attackers exploiting these vulnerabilities, some of the reasons and causes that may contribute to the security gap, and how they can be managed to prevent these attacks from happening. Also, we will explore the growing future trends, like Artificial Intelligence (AI) and automation, that may shape the future of the digital landscape and ultimately play a role in how Windows Security may be influenced.

Research Overview and Resources Utilized

The goal of this research was to explore and dive deeper into commonly found Windows exploits and link them to real-life attacks, potential root causes, strategies to mitigate, and emerging threats. The research will primarily observe:

1. Vulnerabilities and exploits that are commonly found: software misconfigurations, bugs, weakened systems due to out-of-date software, and overall flaws.
2. Real-Life examples: WannaCry, NotPetya, EternalBlue, etc.
3. Source or Root Causes: human/user error, sophisticated systems, patch/update issues, and legacy systems.
4. Mitigation Efforts: network firewall capabilities, tools for endpoint security, management of patches, and overall security posture.
5. Emerging threats: trends of AI and automation open opportunities for new exploits and vulnerabilities, zero-day markets, cloud usage, etc.

This research utilizes aspects from all corners of cybersecurity, including academic literature, reports, and real-life examples to provide a better understanding of Windows Security vulnerabilities and their importance and repercussions.

Frameworks and Methods

Frameworks are structured guidelines on how best to protect, mitigate, and respond to threats. This research pools together different industry-wide established frameworks and other models and tools to gain a better understanding of Windows system vulnerabilities and exploits.

1. The Vulnerability Lifecycle Model

This model establishes the process of how a vulnerability progresses:

1. Discovery – A flaw or exploit is discovered by someone.
2. Disclosure – Vulnerability is reported to the proper chain of command.

3. Patch Development – Microsoft establishes and creates a patch.

4. Patch Deployment – Personnel can update their system with the patch.

5.Exploitation – Attackers take advantage of these exploits before or after the patch.

Understanding this Lifecycle Model helps create a better understanding of why or how these vulnerabilities persist and are taken advantage of.

2. NIST Cybersecurity Framework (NIST CSF)

The NIST CSF is a guideline that highlights the proper steps the organization or user should take to best mitigate threats. The five core functions include Identify, Protect, Detect, Respond, and Recover, and this allows for a proper evaluation of maintaining security posture.

3. Case Study

Attackers have exploited and taken advantage of vulnerabilities. Infamous cases include WannaCry and NotPetya, and analyzing these attacks will help create a better understanding of what went wrong, what failed, and what can and should be done in the case of another similar attack.

These three frameworks and methods will allow us to confirm that our research is based on accepted and supported methodologies industry-wide.

Tools and Results

To better understand how vulnerabilities and exploits are taken advantage of, we need to understand the tools utilized by both the defender and attacker.

1. Windows Security Tools

Windows has certain tools already baked into the system. Examples include Microsoft Defender for Endpoint and Windows Defender Endpoints. These tools provide basic levels of threat detection, scans for malware, and protection for endpoints. Windows Update Services (WSUS) is a centralized patch management system that allows for deployment of patches for known vulnerabilities and exploits. Administrators can detect suspicious login or unusual activities through event viewing and auditing of security. Admins can escalate or de-escalate privileges for better control of who has access.

2. Tools and Techniques Attackers Utilize

Attackers utilize a plethora of Windows tools and other exploitation tools that include those purchased through the Zero-Day Market. Commonly used tools include PowerShell exploitation frameworks like PowerShell Empire, remote code execution exploits like EternalBlue (MS17-010, already patched), harvesting tools like Hash Dump, malware loaders, and phishing toolkits, to name a few. Attackers use these tools to gain an advantage and exploit vulnerabilities.

3. Case Studies Research and Examples

Through our research of former incidents, a commonality can be found in these studies. Examples include organizations using out-of-date systems, bad maintenance of ports (leaving them exposed or open), human error, and overall poor security posture. These examples and findings support the need for properly maintained security hygiene and the need to be actively supporting and updating one’s security system.

Commonly Found Vulnerabilities and Evaluation

1. Code Execution

   Attackers can execute unpredictable code by taking advantage of buffer overflow, bugs pertaining to memory corruption, and unanticipated input flaws. These errors can lead to complete system integrity failure and are very dangerous.

2. Escalation of Privileges

   Attackers begin with the lowest level of privileges and work their way up to gaining administrative services, taking advantage of permission settings being improperly configured and weak user account control (UAC). Privileges are typically the gateway to compromising systems.

3. Authentication

   Due to a lack of strong security hygiene, having weak password rules, poor authentication guidelines, and insecure credential storage, attackers utilize attacks like Pass-the-Hash and Kerberos Golden Ticket to bypass and move throughout the system, causing further compromises.

4. Zero-Day

   These are vulnerabilities and exploits that have been recently discovered that attackers take advantage of before patches are issued. It is commonly found in the Zero-Day market that attackers will utilize them, and they are highly valuable.

Real-Life Occurrences

1. WannaCry

   Many consider this to be one of the deadliest cyberattacks to date. There was a Windows SMBv1 vulnerability known as EternalBlue that was exploited. It was

ransomware that locked users' files and demanded money for the key. Once a system was compromised, it laterally spread throughout the system. Microsoft managed to release a patch, but the damage was already done.

2. NotPetya

   NotPetya was an attack that similarly used the EternalBlue exploit to attack Ukraine. It was essentially malware designed to destroy information and data. This attack crippled companies and caused millions of dollars in damage. Organizations lost their information forever.

These are two examples of infamous attacks that highlight how deadly and impactful exploitation can be.

Root Causes

1. Human/User Error

   Misconfiguration, maintaining weak passwords, not updating systems, and lack of monitoring are vulnerabilities.

2. Sophisticated Systems

   Windows is massive software, and the chance of there being an exploit or vulnerability is high due to its size.

3. Patch/Update Issues

   Organizations and companies delay updating software because of concerns that updates may stall or affect day-to-day and big-picture operations. This delay makes systems vulnerable.

4. Legacy Systems

   Windows is constantly updated and improved, but some organizations, whether due to cost or other reasons, choose not to update to the latest software. Legacy software typically does not receive patches and is no longer supported by Microsoft after a certain period, leaving it vulnerable to attacks.

Mitigation Efforts

1. Network Security

   Windows comes with a built-in firewall, but organizations need to properly maintain and keep it up to date and allow proper configurations to be implemented, as well as IDS/IPS systems.

2. Endpoint Detection and Response

   Microsoft Defender for Endpoint uses automation and behavioral analysis to identify suspicious activities.

3. Management of Patches

   Updates can be maintained and utilized through tools like WSUS, and vulnerabilities that are critical should be prioritized.

4. Security Posture

   Creating a culture in the work environment to prevent human error, along with security awareness training, needs to be spread and utilized.

Emerging Threats

Windows Security is constantly evolving and adapting to the digital landscape. AI has become the forefront of technology, and attackers are utilizing it to exploit vulnerabilities

and create tools, like malware, to gain an advantage. Windows Defender is reciprocating by utilizing AI to mitigate these attacks; however, this creates another layer of sophistication, as it becomes a battle of resources that might not play out well for smaller and independent organizations. Other potential threats include the usage of a hybrid system that utilizes the cloud and local infrastructure, which opens gateways for attackers to find new methods to attack, and the growth of the Zero-Day Markets. Technology is rapidly improving, but at the same time, creating new exploits and vulnerabilities, if not properly addressed or maintained, can be taken advantage of.

Conclusion

Windows Security continues to grow and develop and remains popular and widely used throughout the world. However, this popularity, as well as the sophistication, causes the continued targeting by attackers. Infamous events like WannaCry and NotPetya highlight the impact exploiting a vulnerability can cause to systems. Root causes of vulnerabilities, such as human error and patching delays, showcase that regardless of Microsoft's investments and utilization of resources, attacks continue to persist. The mitigation of attacks like these prescribes the necessity for a simple but effective strategy that highlights the integration of methods mentioned, like security awareness, endpoint security, and system maintenance. Cybersecurity never sleeps, as threats continue to emerge, with examples including the popularity of Artificial Intelligence and other growing exploits and vulnerabilities. Ultimately, utilizing past events, as well as the current methodologies in the current landscape, will need to be integrated to maintain and improve Windows Security.

References

Haider, W., Creech, G., Xie, Y., & Hu, J. (2016). Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks. *Future Internet, 8*(3), 29. https://doi.org/10.3390/fi8030029

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences, 80*(5), 973–993. https://doi.org/10.1016/j.jcss.2014.02.005

Kshetri, N., & Voas, J. (2017). WannaCry ransomware attack: A wake-up call. *Computer, 50*(7), 91–95. https://doi.org/10.1109/MC.2017.201

Kulshrestha, A., Song, G., & Zhu, T. (2023). *The inner workings of Windows security*. *arXiv*. https://arxiv.org/abs/2312.15150

Li, Y., Sun, L., & Xu, T. (2022). AI-driven malware and the future of cybersecurity. *ACM Computing Surveys, 54*(8), Article 183. https://doi.org/10.1145/3533339

National Institute of Standards and Technology. (n.d.). *Cybersecurity Framework.* Retrieved November 27, 2025, from https://www.nist.gov/cyberframework

Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS).* NIST Special Publication. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50951