A Survey on Windows Login Bypass Techniques: A Red Teamer's Perspective

First Author[#1], Second Author[*2], Third Author[#3]

[#]*First-Third Department, First-Third University*
Address Including Country Name
[1]first.author@first-third.edu
[3]third.author@first-third.edu

[*]*Second Company*
Address Including Country Name
[2]second.author@second.com

Abstract— **The security of the user authentication system in Microsoft Windows is a fundamental element of contemporary digital defense. Nonetheless, for penetration testers, red teams, and digital forensic specialists, circumventing this security barrier is an essential goal. This paper offers a thorough overview of the strategies and tools employed to execute Windows login bypass. We examine the progression of these techniques, ranging from classic offline attacks aimed at the SAM (Security Account Manager) databaseto advanced "live" assaults that take advantage of hardware interfaces and flaws in modern authentication systems like Windows Hello. The main conclusion of this survey is that the multilayered, hardware-integrated security framework of the latest Windows 10 and 11—including Secure Boot, the TPM (Trusted Platform Module), and Virtualization-Based Security (VBS)—has made most outdated bypass techniques ineffective. As a result, the attack surface has transitioned to particular, patchable weaknesses in live system elements and protocols. This paper presents a categorized overview of these attack vectors, assesses the efficacy of related tools, and examines the challenges and future research avenues, including the rising threat posed by AI-driven autonomous attack agents.**

Keywords— **Include at least 5 keywords or phrases**

I. INTRODUCTION

A. Background

The Microsoft Windows operating system serves as a fundamental platform for both business and personal computing. At its foundation, the user authentication system functions as the main security barrier, safeguarding sensitive information from unauthorized access. This system has undergone substantial changes since the password-based approaches used in earlier versions of Windows to the advanced multi-factor authentication methods available in Windows 10 and 11. Contemporary authentication increasingly focuses on password-less approaches like Windows Hello, which utilizes biometrics and hardware-based security elements such as the TPM (Trusted Platform Module). These innovations are intended to address a continually evolving and more intricate threat landscape.

B. Problem Statement and Motivation

From a defensive standpoint, the login interface serves as the most vital barrier safeguarding a system's information, particularly when supported by Full-Disk Encryption (FDE) like BitLocker. Nevertheless, for offensive security experts and digital forensic analysts, the primary goal is to access an operational system without modifying its state. The transient memory (RAM) of an active machine holds crucial information, including current processes, session tokens, and cryptographic keys, which are lost when the system is powered down.

Conventional login bypass tools, usually included in recovery suites like Hiren's BootCD, generally function through "destructive" methods—they reset, erase, or create user passwords by directly altering the SAM (Security Account Manager) database offline. This approach is frequently considered inappropriate in red team activities or forensic examinations, as it modifies evidence, notifies defenders of a breach, and may render user-encrypted data unreachable. This situation has driven the need for "intact" or "non-destructive" bypass techniques that facilitate access while maintaining the original user credentials. Specific tools, such as the "Windows Login Unlocker Pro PE," assert they can provide this capability, enabling an attacker to log in without the original password. However, the reported ineffectiveness of this tool against more recent Windows versions

(post-22H2) underscores a significant gap and drives this investigation: to document the latest developments in Windows login bypass techniques and to analyze why older methods are ineffective against today's fortified systems.

C. Scope and Objectives

The main goal of this paper is to present an extensive overview of the methods and tools used to circumvent the Windows login process. The focus is on attacks that require local physical access and categorizes these techniques according to their effects on system integrity. This review will investigate a wide range of methods, including traditional registry and SAM file manipulation, contemporary hardware-based exploits that utilize Direct Memory Access (DMA) or malicious peripherals, and the exploitation of weaknesses within the Windows Hello framework. An important aim is to assess the efficacy of these methods against the security framework of fully-updated Windows 10 and 11 systems.

D. Paper Structure

The rest of this paper is structured as follows. Section II covers the essential elements of the Windows security framework related to authentication. Section III provides a classification of techniques for bypassing login mechanisms and the tools used to execute them. Section IV examines the security environment of contemporary Windows and discusses how to address outdated vulnerabilities. Section V explores existing challenges and potential avenues for future research, especially concerning the impact of artificial intelligence on automating attacks. Lastly, Section VI offers a conclusion that encapsulates the main insights from this review.

II. FOUNDATIONAL WINDOWS SECURITY ARCHITECTURE

Grasping Windows login bypass techniques requires a basic comprehension of the security elements that these approaches aim at. The Windows authentication framework is an intricate, multi-tiered system intended to safeguard user credentials and maintain system integrity. This part outlines the essential elements that play a role in the authentication process, ranging from outdated database files to contemporary hardware-based protections.

A. The Security Account Manager (SAM) Database

The Security Account Manager (SAM) is a crucial database file located in Microsoft Windows operating systems that stores local user accounts and their associated password hashes. When a user attempts to log in locally (not through a domain), the entered credentials are processed and matched against the data stored in this database. This file can be found in the system registry at the location ‘%systemroot%\system32\config\sam’.

The SAM file, owing to its sensitive characteristics, is safeguarded by a file-locking mechanism that the Windows kernel applies during standard operation, making unauthorized copying or direct alterations impossible [2]. This essential protection gave rise to "offline attacks," which bypass the kernel lock by booting from external media to access the file system directly. Programs like Windows Login Unlocker Pro typically operate by creating bootable media to execute these types of offline modifications.

Historically, passwords in the SAM database have been stored using two formats: LAN Manager (LM) hash and NT LAN Manager (NTLM) hash [2]. LM hashes are regarded as cryptographically weak and greatly vulnerable to brute-force attacks, leading to their use being disabled by default in newer versions of Windows. To enhance the protection of the SAM database against offline attacks, Microsoft implemented the System Key (SYSKEY), a tool that encrypts the database using a 128-bit key [2].

B. The Interactive Logon Process and LSASS

The interactive logon for Windows is managed by a series of specialized processes. The Windows Logon process (winlogon.exe) starts the session and executes the Logon User Interface (LogonUI.exe), which offers the visual interface for users to input

their credentials, including passwords, PINs, or biometric information.

After credentials are entered, they are forwarded to the Local Security Authority Subsystem Service (LSASS), a vital process (lsass.exe) that acts as the backbone of Windows authentication. LSASS is tasked with confirming the user's identity through various authentication packages. When authenticating local accounts, LSASS uses the MSV1_0 package to check the supplied credentials against the SAM database. For machines connected to a corporate network, LSASS generally relies on the Kerberos security support provider (SSP) to authenticate the user with a remote domain controller. Once validation is successful, LSASS generates the user's access token, which includes their security identifiers (SIDs) and privilege levels, and then launches the user's shell (explorer.exe). Given that LSASS manages sensitive information, such as credential hashes held in memory, it is often targeted by sophisticated credential theft methods, including memory dumping [2].

C. Full-Disk Encryption and Modern Hardware-Based Security

The risk of offline assaults on the SAM database is significantly reduced through the use of Full-Disk Encryption (FDE). Microsoft's built-in FDE option, BitLocker, secures the complete operating system volume, making the SAM file and all other system information inaccessible without the correct decryption key. This redirects the attacker's attention from offline file alterations to circumventing the authentication processes of an active, operational system.

In order to safeguard the BitLocker decryption key, contemporary Windows systems utilize advanced hardware security functions extensively.

1) *Trusted Platform Module (TPM):* The TPM is a security processor based on hardware that is a standard element in contemporary computers. BitLocker utilizes the TPM to safely keep the disk encryption key, a technique intended to provide greater security than simply placing the key on the disk. The TPM is designed to provide the decryption key only when it confirms that the boot process remains unchanged.

2) *Secure Boot and the Chain of Trust:* The boot process's integrity is maintained through a "chain of trust." Starting with the Unified Extensible Firmware Interface (UEFI), each element in the boot sequence—from the bootloader to the OS kernel—is cryptographically assessed, and its signature is logged in the TPM's PCRs (Platform Configuration Registers). If an attacker modifies any segment of this chain (for example, by attempting to boot from a compromised USB drive), the PCR values will not match the expected state, leading the TPM to refuse to unseal the BitLocker key, thus preventing the system from starting and protecting the encrypted data.

3) *Virtualization-Based Security (VBS):* Modern iterations of Windows, specifically Windows 10 and 11, utilize Virtualization-Based Security (VBS) to enhance the protection of essential components such as LSASS. VBS leverages hardware virtualization for a secure, isolated memory area where sensitive processes are able to function, shielding them from threats that may arise from a breached kernel.

4) *Kernel DMA Protection:* To protect against advanced hardware attacks that involve accessing system memory directly, Windows has introduced Kernel DMA (Direct Memory Access) Protection. This capability blocks unauthorized peripherals (for instance, those connected through Thunderbolt) from obtaining direct access to RAM, reducing the risk of a category of attacks that have previously been employed to extract encryption keys and other confidential information.

These multiple layers of defense have made many conventional bypass techniques outdated on properly set up, contemporary systems, compelling both attackers and security researchers to search for weaknesses within the active authentication protocols.

III. A TAXONOMY OF WINDOWS LOGIN BYPASS TECHNIQUES

Methods for bypassing Windows login have advanced as the operating system has become more secure. These techniques vary from straightforward offline file changes to intricate hardware-dependent and protocol-based attacks. This section offers a classification of these approaches, organizing them according to their operational states and examining their mechanisms, consequences, and the tools that execute them.

A. Classification by System State

Bypass methods can generally be divided into two primary categories depending on the condition of the target system throughout the attack.

1) *Offline Attacks*: These are the most classical and well-documented techniques. An offline attack necessitates that the attacker power down the target computer and start it using a different, attacker-controlled operating system, usually from a USB drive or live CD. This provides the attacker with direct and unrestricted access to the Windows file system, as the native Windows kernel and its security measures are not operational. Methods such as direct SAM file editing and manipulation of the registry fit within this category.

2) *Online and "Live" Attacks*: These attacks occur when the Windows OS is operational or in a pre-boot state with the hardware engaged. Rather than booting into a different OS, they take advantage of weaknesses in active processes, hardware connections, or authentication systems. For instance, assaults directed at a locked but operational computer to retrieve credentials from memory or via malicious peripherals illustrate this tactic. Such methods are often favored in red team exercises as they maintain the current state of the system, including the information stored in RAM.

B. Analysis of Common Bypass Methods

The following is a detailed examination of particular bypass methods, evolving from traditional destructive techniques to contemporary, frequently non-intrusive exploits.

1) *Registry Manipulation (The "Sticky Keys" Method)*: This well-known offline attack entails modifying the Windows registry to obtain elevated access directly from the login interface. The attacker starts the machine using an external drive, loads the registry hives of the system, and edits a key to substitute the executable of an accessibility feature (such as sethc.exe for Sticky Keys or Utilman.exe for Utility Manager) with the Command Prompt (cmd.exe) [Study on Security Auditing of Windows Registry Database]. When the accessibility icon is activated on the login screen, a command prompt with SYSTEM-level permissions is opened instead of the original tool. From this point, the attacker can run commands to either create a new administrator account or alter the password of an existing user using net user. This technique is deemed destructive as it modifies the condition of the system and can be easily detected.

2) *Direct SAM File Modification*: This is another basic offline attack method. By utilizing a bootable Linux environment, an attacker can employ tools such as ‘chntpw’ to directly read and alter the SAM database file. This enables several invasive actions: resetting a user's password, elevating a standard user account to an administrator level, or unlocking a disabled account [2]. This approach is very

effective on systems that do not utilize disk encryption, but it is fundamentally destructive as it permanently changes the credential database.

3) *Hardware-Based Attacks (DMA and Rogue Devices)*: These are more advanced "live" attacks that target the system's hardware interfaces.
    - **Direct Memory Access (DMA) Attacks:** An attacker with physical access can use a malicious peripheral connected via a port that allows Direct Memory Access (e.g., Thunderbolt, Firewire) to read or write directly to the system's RAM. This can be utilized to inject malicious code or to dump the memory of lsass.exe, which contains credential hashes. However, modern Windows systems mitigate this threat with Kernel DMA Protection, which uses the Input/Output Memory Management Unit (IOMMU) to block unauthorized DMA access.
    - **Rogue USB Devices:** Devices like the LAN Turtle are designed to exploit how Windows handles network connections from a locked state. The LAN Turtle, an Ethernet-over-USB adapter, can be plugged into a locked machine. It acts as a unauthorized DHCP server and spoofs network traffic, tricking the workstation into sending its NTLMv2 password hash in an authentication attempt. The attacker captures this hash and can crack it offline to reveal the user's password. This method can be effective but requires specific network conditions.
4) *Vulnerabilities in Modern Authentication (Windows Hello)*: Despite being more secure than passwords, the Windows Hello framework has been the subject of several vulnerability disclosures.
    - **Migration Attack on TPM-less Devices:** On systems lacking hardware protection (i.e., no TPM), the authentication data for Windows Hello is not sufficiently protected. Researchers have demonstrated a "migration attack" where this data can be retrieved from a device, decrypted, and then transferred to an attacker's machine. This allows the attacker to impersonate the victim and access their Microsoft online accounts and services, even bypassing two-factor authentication.
    - **Biometric Spoofing via USB:** For facial recognition, it has been demonstrated that an attacker with physical access can use a custom USB device to masquerade as the legitimate infrared camera. By capturing or reproducing a suitable IR image of the victim, this rogue device can feed the spoofed data to the Windows Hello service, successfully bypassing the facial authentication check.
    - **Biometric Database Tampering:** Recent research revealed that an attacker who has already achieved local administrator privileges can tamper with the biometric database used by Windows Hello. This allows them to register their own biometric data (e.g., their own face) to the victim's account, enabling them to log in through Windows Hello.

C. Survey of Bypass Tools

The techniques described above are implemented in various publicly available and specialized tools.

1) *Recovery and Forensic Suites (e.g., Hiren's BootCD)*: These are bootable toolkits that bundle a wide array of system recovery and security utilities. For password bypass, they typically include open-source tools like ‘chntpw’ that perform direct SAM file modification. As discussed, these tools are effective for recovery but are considered destructive and "loud" from a red team perspective, as their use is easily detectable and alters the target system's credentials.

2) *Specialized Bypass Tools (Windows Login Unlocker Pro PE)*: In contrast to general recovery suites, specialized commercial tools claim to offer more advanced, non-destructive bypass capabilities. Your research identified "Windows Login Unlocker Pro PE," which purports to install a bypass mechanism that allows logging into an account without entering a password at all, leaving the original password intact. Based on your findings, the effectiveness of this specific tool is limited to older Windows versions (prior to Windows 10 22H2). This strongly suggests that the underlying vulnerability or technique it exploits has been patched or mitigated by the architectural security improvements introduced in modern Windows 10 and 11, which will be discussed in the next section.

IV. THE MODERN WINDOWS SECURITY LANDSCAPE (WINDOWS 10 22H2+ AND WINDOWS 11)

The Windows security model has undergone a radical transformation over the last decade. While older versions of Windows were vulnerable to a wide range of straightforward bypass techniques, modern iterations—particularly Windows 10 (version 22H2 and later) and Windows 11—incorporate a layered, hardware-integrated defence-in-depth strategy. This has rendered many legacy attack vectors obsolete and significantly raised the complexity for attackers.

A. The Mitigation of Offline Attacks

As discussed in Section III, offline attacks that involve booting from external media were once the most reliable method for bypassing Windows login. The effectiveness of this entire class of attack has been nullified on properly configured modern systems by the synergistic combination of three key technologies:

1) *UEFI Secure Boot:* This is a firmware-level security standard that ensures the device boots only using software that is trusted by the Original Equipment Manufacturer (OEM). It creates a "chain of trust" by validating the cryptographic signature of each piece of boot software, from the firmware drivers to the OS bootloader. This prevents an attacker from simply booting the machine from an unauthorized USB drive containing bypass tools.

2) *Trusted Platform Module (TPM):* The TPM provides a hardware root of trust. When used with BitLocker, the TPM stores the disk encryption key and is configured to release it only if the boot process is unmodified, as measured by the Platform Configuration Registers (PCRs). Any attempt to tamper with the bootloader or boot from an external device will alter the PCR values, causing the TPM to withhold the decryption key.

3) *BitLocker Full-Disk Encryption (FDE):* With the disk fully encrypted, an attacker who manages to bypass Secure Boot and boot from an external OS will be unable to read or modify any files on the Windows volume, including the SAM database or the registry hives. The data is cryptographically inaccessible without the key, which is protected by the TPM.

Together, these features effectively close the door on classic offline manipulation, forcing adversaries to find vulnerabilities in the live, running system.

B. Hardening Measures in the Live Environment

Recognizing the shift in attack vectors, Microsoft has invested heavily in protecting the live system environment, even when a user is not logged in.

1) *Virtualization-Based Security (VBS) and Credential Guard:* VBS leverages hardware virtualization to create an isolated and protected region of memory. A key feature that uses VBS is **Credential Guard**, which runs the LSASS process within this secure, virtualized environment. This prevents even a compromised kernel from directly accessing the memory of LSASS, thereby mitigating the credential-dumping attacks that aim to extract password hashes from a running system. This is a significant hardening measure against tools that rely on memory scraping techniques.

2) *Kernel DMA Protection:* To counter sophisticated hardware attacks, modern Windows systems implement Kernel DMA Protection. This feature utilizes the IOMMU (Input-Output Memory Management Unit) to block unauthorized peripherals from performing Direct Memory Access (DMA) attacks, which could otherwise be used to read the contents of system RAM, including sensitive credential data. This directly mitigates the threat posed by malicious devices connected via Thunderbolt or other DMA-capable ports.

3) *Continuous Patching and Vulnerability Response:* The Windows ecosystem is subject to constant security analysis by researchers worldwide. When vulnerabilities are discovered, Microsoft typically responds with security patches delivered via Windows Update. A prime example is the response to the Windows Hello biometric spoofing attack. After researchers demonstrated that a rogue USB camera could be used to bypass facial recognition, Microsoft issued a patch (addressing CVE-2021-34466) which introduced the concept of a "secure camera" protocol, ensuring that the OS only trusts input from specific, certified hardware. This ongoing cycle of discovery and mitigation is a primary reason why bypass tools have a limited shelf life.

C. The Consequence: Obsolescence of Legacy Bypass Tools

The culmination of these architectural improvements explains why many of the tools and techniques that were effective in the past are no longer viable.

1) Tools that rely on **offline SAM modification** (e.g., ‘chntpw’ in Hiren's BootCD) are defeated by BitLocker.

2) Hardware tools that rely on **DMA attacks** are blocked by Kernel DMA Protection.

3) Your own research finding-that **Windows Login Unlocker Pro PE** is ineffective on Windows 10 versions post-22H2-is a direct consequence of this security hardening. While the exact mechanism of the tool is not publicly documented, its failure suggests it exploited a specific software vulnerability within the Windows logon process (e.g., in winlogon.exe, LogonUI.exe, or an associated LSASS authentication package). Such a flaw would have been identified and patched by Microsoft in one of the cumulative updates leading up to and included in the 22H2 release.

Therefore, the modern Windows security landscape is not defined by a single defensive feature but by a deeply integrated, multi-layered system that makes a generic, one-size-fits-all "live login bypass" a far more challenging and often infeasible endeavour.

V. DISCUSSION: CHALLENGES AND FUTURE RESEARCH DIRECTIONS

The evolution of Windows security from a relatively simple password-gate to a hardware-integrated, multi-layered defence system has fundamentally reshaped the landscape of login bypass attacks. The survey of techniques reveals a clear trend: as Microsoft hardens the core operating system, the focus of sophisticated attackers and security researchers has shifted from simple, predictable exploits to complex, niche vulnerabilities and automated attack frameworks.

A. The "Intact Bypass" Challenge on Modern Systems

The central challenge for any red teamer or attacker is achieving an "intact bypass" on a modern, fully-patched Windows 11 system. As demonstrated in Section IV, the foundational offline methods are largely obsolete due to the combined defences of Secure Boot, TPM, and BitLocker. This forces attackers into the live environment, where they are met with formidable defences like VBS-protected Credential Guard and Kernel DMA Protection.

This security posture means that a generic, universally effective login bypass tool is likely a thing of the past. The failure of older tools like Windows Login Unlocker Pro PE on modern systems underscores this reality. Success is now predicated on discovering and exploiting zero-day or other specific, unpatched vulnerabilities in the complex web of software and hardware that constitutes the authentication process. These exploits are, by their nature, transient and are quickly rendered ineffective by security patches.

B. Future Research Directions and Emerging Attack Vectors

The academic literature points to several key areas where future research and attack methodologies are likely to concentrate.

1) *Offensive Artificial Intelligence and Automated Exploitation*: The manual process of probing for vulnerabilities and chaining exploits is time-consuming. Future attacks are likely to be driven by artificial intelligence. Research has already demonstrated the feasibility of training a deep reinforcement learning agent to automate the task of local privilege escalation. Such an agent can learn to identify system misconfigurations (e.g., hijackable DLLs, unquoted service paths) and execute the optimal sequence of actions to elevate its privileges far more rapidly and adaptively than a human operator or a static script. This represents a paradigm shift from manually executed techniques to autonomous agents capable of performing red team operations.

2) *The Biometric Attack Surface*: Windows Hello, while a significant step up from password-based authentication, has introduced a new and complex attack surface. As demonstrated by the "Windows Hell No" vulnerability and research into biometric spoofing via rogue USB cameras, the implementation of biometric systems is fraught with potential weaknesses. While Microsoft has patched specific flaws, future research will likely focus on:
    1. Side-channel attacks against biometric sensors.
    2. Exploiting the enrolment and database management processes.
    3. Finding new ways to inject spoofed biometric data that bypass hardware-trust checks.

3) *Protocol-Level Weaknesses*: Even when the OS components themselves are secure, the underlying authentication protocols can have design flaws. The 2015 research that demonstrated a bypass of BitLocker did not attack BitLocker itself, but rather a weakness in the Kerberos password reset protocol that allowed an attacker to poison the cached credentials of a domain-joined machine. This highlights that complex protocols like Kerberos and NTLM will remain a fertile ground for security research, with the potential for novel attacks that circumvent OS-level protections.

C. Implications for Defensive Security

This survey of bypass techniques serves as a crucial resource for defensive "blue teams" and system administrators. The key takeaway is that a robust defence relies on a proactive and layered security posture.

1) *Configuration is Key:* The effectiveness of modern Windows defences is contingent upon their proper configuration. Administrators must ensure that Secure Boot, TPM, BitLocker, and Credential Guard are enabled and correctly configured across their enterprise fleet.
2) *Detection and Auditing:* Since a successful bypass often indicates a specific, unpatched vulnerability, robust system auditing is critical. The Windows Registry and Event Logs contain a treasure trove of artifacts that can indicate a failed or successful attack [Study on Security Auditing of Windows Registry Database]. Monitoring for unusual registry modifications, anomalous login events, or unexpected hardware enumeration can provide early warnings of a compromise.

3) *The AI Arms Race:* The rise of offensive AI necessitates the development of AI-driven defensive systems. Several researchers have proposed that future security solutions should employ machine learning to analyse system and command behaviour in real-time. An AI-powered defence could, for example, analyse a sequence of PowerShell commands *before* execution to determine if they align with a known attack pattern, thereby proactively blocking the threat,. This points to an inevitable "arms race" where autonomous defensive agents will be needed to counter autonomous attackers.

In summary, the field of Windows login bypass is far from static. While the bar for a successful attack has been raised considerably, new frontiers in AI, biometrics, and protocol analysis are continuously emerging, ensuring that this cat-and-mouse game between attackers and defenders will persist for the foreseeable future.

VI. Conclusions

This survey has charted the evolution of Windows login bypass techniques, analyzing the interplay between offensive methodologies and defensive innovations. Our investigation began by examining the foundational, "destructive" offline attacks that targeted the SAM (Security Account Manager) databaseand the Windows Registry. For years, these methods, implemented in widely available tools, provided a reliable means of gaining access to systems that lacked full-disk encryption.

However, the primary finding of this paper is that the security landscape has fundamentally and decisively shifted. The architectural hardening of modern Windows operating systems—specifically through the integrated, hardware-rooted defenses of UEFI Secure Boot, the TPM (Trusted Platform Module), BitLocker, and Virtualization-Based Security (VBS)—has rendered this entire class of traditional offline attacks largely obsolete on properly configured systems.

This has resulted in a clear migration of the attack surface. The focus of sophisticated adversaries and security researchers is no longer on the offline file system but on the live, running system. As this survey has detailed, modern bypass techniques are now predicated on discovering and exploiting specific, often transient, vulnerabilities within complex components like the Windows Hello biometric framework, associated hardware interfaces, and the underlying authentication protocols themselves.

Consequently, this paper confirms the initial research hypothesis: a universal, non-destructive ("intact") login bypass tool for a fully-patched, modern Windows 11 system is no longer a trivial or likely proposition. Specialized tools that claim such capabilities, like the Windows Login Unlocker Pro PE, are inevitably built upon specific, patchable flaws, explaining their limited lifespan and ineffectiveness against newer OS versions. The future of this domain does not lie in a single "magic bullet" but in the realm of automation and artificial intelligence, where autonomous agents will be developed to discover and chain exploits in real-time. This signals the beginning of a new, more complex chapter in the ongoing security arms race between attackers and defenders in the Windows ecosystem.

VII. References

[1] J. V. A. Ribeiro and D. M. Caldas, "Survey on the possibility of Windows 10 live login bypass," Brazilian Journal of Development, vol. 8, no. 3, pp. 17905-17916, Mar. 2022.

[2] "Introduction to Security Accounts Manager (SAM) Database," ScienceDirect. [Online]. Available: ScienceDirect, SAM overview [Online]

[3] I. Haken, "Bypassing Local Windows Authentication to Defeat Full Disk Encryption," in Black Hat Europe 2015, Amsterdam, Netherlands, 2015. [Online]. Available: https://www.blackhat.com/docs/eu-15/materials/eu-15-Haken-Bypassing-Local-Windows-Authentication-To-Defeat-Full-Disk-Encryption-wp.pdf.

[4] E. Kim and H.-K. Choi, "Security Analysis and Bypass User Authentication Bound to Device of Windows Hello in the Wild," Security and Communication Networks, vol. 2021, Art. ID 6245306, pp. 1-13, 2021.

[5] J. Tashi, "Study on Security Auditing of Windows Registry Database," IJSTE - International Journal of Science Technology & Engineering, vol. 8, no. 1, pp. 1-5, Jul. 2021.

[6] N. Mohamed, "Study of bypassing Microsoft Windows Security using the MITRE CALDERA Framework," F1000Research, vol. 11, no. 422, 2022.

[7] K. Kujanpää, W. Victor, and A. Ilin, "Automating Privilege Escalation with Deep Reinforcement Learning," in Proc. 14th ACM Workshop on Artificial Intelligence and Security (AISec '21), Virtual Event, Republic of Korea, 2021, pp. 1-12.

[8] "Researchers reveal 'Windows Hell No' vulnerability in Windows Hello biometric system," IDTechWire, Aug. 2025. [Online]. Available: https://idtechwire.com/researchers-reveal-windows-hell-no-vulnerability-in-windows-hello-biometric-system/.

[9] O. Tsarfati, "Bypassing Windows Hello without Masks or Plastic Surgery," CyberArk Threat Research Blog, 2021. [Online]. Available: https://www.cyberark.com/resources/threat-research-blog/bypassing-windows-hello-without-masks-or-plastic-surgery.

[10] A. Kumar and S. K. Shrivastava, "A Comprehensive Study on Windows Password Vulnerabilities," Indian Journal of Computer Science, vol. X, no. 4, 2016.

[11] "Windows Login Unlocker Pro," KaranPC. [Online]. Available: https://karanpc.com/windows-login-unlocker-pro-download/.

[12] "Windows Login Unlocker Pro PE 1.8 (x86/x64) Bootable," FC Portables. [Online]. Available: https://www.fcportables.com/windows-login-unlocker-boot/.

[13] Microsoft, "Windows authentication overview," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-overview.

[14] Microsoft, "Windows Authentication Concepts," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-concepts.

[15] Microsoft, "KB5005478: Enhanced sign-in security for Windows Hello," Microsoft Support. [Online]. Available: https://support.microsoft.com/en-gb/topic/kb5005478-windows-hello-cve-2021-34466-6ef266bb-c68a-4083-aed6-31d7d9ec390e.

[16] A. Neyaz and N. Shashidhar, "USB Artifact Analysis Using Windows Event Viewer, Registry and File System Logs," Electronics, vol. 8, no. 11, p. 1322, 2019.

[17] S. Zargari and J. Dyson, "Memory forensics: comparing the correctness of memory captures from locked Windows 10 machines using different boot capture vectors," Latin-American Journal of Computing, vol. 9, no. 2, pp. 37-51, 2022.

[18] S. Sivakorn, I. Polakis, and A. D. Keromytis, "The PRMitM Attack: Application-level Man-in-the-Middle on Password Reset," in 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 553-568.

[19] T. M. H. F. A. Al-Ameen and L. A. Al-Khattat, "AI-Based Authentication Systems: A Review," arXiv preprint arXiv:2312.15150, 2023.

[20] A. M. D. R. Chowdhury, "A systematic review of Windows forensics: From 2010 to 2020," Cybersecurity, vol. 7, no. 1, 2021.

[21] J. P. Biggs and C. Williams, "Biometric Authentication: A Review," in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 2019.

[22] National Cyber Security Centre (NCSC), "Using biometrics," Device Security Guidance, 2024. [Online]. Available: https://www.ncsc.gov.uk/collection/device-security-guidance/policies-and-settings/using-biometrics.

[23] T. Webs, "Download Windows Login Unlocker 1.6," taiwebs.com. [Online]. Available: https://en.taiwebs.com/windows/download-windows-login-unlocker-8031.html.

[24] "Windows Password Unlocker Professional," software.informer.com. [Online]. Available: https://windows-password-unlocker-professional.software.informer.com.

[25] Passcape, "Resetting a Windows password," www.top-password.com. [Online]. Available: https://www.top-password.com/knowledge/unlock-windows-password.html.

[26] A. Kumar, "Top 10+ Best Windows Password Remover / Cracker Tools 2024," ruhanirabin.com. [Online]. Available: https://www.ruhanirabin.com/top-best-windows-password-remover/.

[27] J. Kozy, "Microsoft Windows RDP Network Level Authentication Bypass (CVE-2019-9510): What You Need to Know," Rapid7 Blog, 2019. [Online]. Available: https://www.rapid7.com/blog/post/2019/06/05/microsoft-windows-rdp-network-level-authentication-bypass-cve-2019-9510-what-you-need-to-know/.