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 user authentication system in Microsoft Windows is a fundamental component of modern digital defence. For penetration testers, red teams, and digital forensic specialists, however, circumventing this security boundary is an essential objective. This paper offers a thorough overview of the strategies and tools employed to execute a Windows login bypass. We examine the progression of these techniques, from classic offline attacks targeting the Security Account Manager (SAM) database to advanced "live" assaults that exploit hardware interfaces and flaws in modern authentication systems like Windows Hello. A central observation of this survey is that the multi-layered, hardware-integrated security architecture of modern Windows 10 and 11—including Secure Boot, the Trusted Platform Module (TPM), and Virtualization-Based Security (VBS)—has rendered most legacy bypass techniques ineffective. As a result, the attack surface has shifted to specific, patchable vulnerabilities in live system components 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— **Windows Security, Login Bypass, Red Teaming, Penetration Testing, SAM Database, Windows Hello, Credential Theft**

I. Introduction

A. The Evolving Battlefield of Windows Authentication

The Microsoft Windows operating system serves as a foundational platform for both business and personal computing. At its core, the user authentication system functions as the primary security boundary, safeguarding sensitive information from unauthorized access. This system has undergone a substantial evolution from the password-based approaches of earlier versions to the advanced multi-factor authentication methods available in Windows 10 and 11. Contemporary authentication increasingly relies on password less strategies like Windows Hello, which utilizes biometrics and hardware-anchored security elements such as the Trusted Platform Module (TPM). These innovations are designed to address a continuously evolving and more sophisticated threat landscape.

B. Problem Statement and Motivation

From a defensive standpoint, the login interface serves as the most critical security barrier for a system, particularly when supported by Full-Disk Encryption (FDE) like BitLocker. For offensive security experts and digital forensic analysts, however, the primary objective is to access a running operational system without modifying its state. The volatile memory (RAM) of a live machine contains crucial information—including running processes, session tokens, and cryptographic keys—that is permanently lost when the machine is turned off.

This requirement is in direct conflict with the methods of conventional login bypass tools. Recovery suites, such as Hiren's BootCD, have historically functioned through "destructive" methods. This approach requires booting from external media to directly alter the Security Account Manager (SAM) database offline, which allows for the resetting, blanking, or creation of user passwords. Such a modification is often unacceptable in red team engagements or forensic examinations because it alters evidence, alerts defenders to a compromise, and can render user-keyed encrypted data inaccessible.

This situation has driven the need for "intact" or "non-destructive" bypass techniques that facilitate access while maintaining the original user

credentials. The existence of specialized tools, like "Windows Login Unlocker Pro PE," which claimed to provide this capability, suggested such methods were feasible. However, the reported ineffectiveness of this tool against modern Windows versions (post-22H2) underscores a significant capabilities gap. This motivates our investigation: to document the latest developments in Windows login bypass techniques and to analyse why older methods are ineffective against today's hardened systems.

II. FOUNDATIONAL WINDOWS SECURITY ARCHITECTURE

To understand how Windows login can be bypassed, one must first understand how a legitimate login works. When a user enters a password, PIN, or uses a biometric scanner, they initiate a complex, multi-stage verification process handled by several critical components working in concert. This process involves both live, in-memory validation and checks against stored credential data, all while being protected by underlying physical and hardware-level security.

This section will deconstruct that process by examining its four primary architectural pillars. We will begin with the Local Security Authority Subsystem Service (LSASS), the core process that orchestrates the live authentication attempt. We will then discuss the Security Account Manager (SAM) database, the historical ledger against which local credentials are verified. Next, we will analyse how Full-Disk Encryption (FDE), specifically BitLocker, provides a fundamental layer of protection against offline attacks that target these components. Finally, we will detail how the security of the entire system—both at boot and during runtime—is anchored in hardware through the Trusted Platform Module (TPM), UEFI Secure Boot, Virtualization-Based Security (VBS), and Kernel DMA Protection.

These components—LSASS, the SAM, BitLocker, and the TPM—represent the principal layers of defence. Consequently, they have become the primary targets for attackers. The following subsections will detail the role and function of each of these elements, establishing the necessary context for the bypass techniques discussed later in this paper.

A. The Core Orchestrator: LSASS and the Logon Process

The live authentication process in Windows is orchestrated by the Local Security Authority Subsystem Service (LSASS), which runs as the critical process lsass.exe. When a user provides their credentials via the Logon User Interface (LogonUI.exe), it is LSASS that receives this information for validation. LSASS is responsible for calling upon various authentication packages to verify the user's identity. For a machine that is part of a corporate network, LSASS typically employs the Kerberos protocol to securely communicate with a domain controller to have credentials verified. For a standalone, local machine, the task falls to a different authentication package. Because LSASS handles sensitive data like credential hashes directly in memory during this process, it is a high-value target for advanced attacks, such as memory dumping, which attempt to steal login information from the system while it is running.

B. The Credential Store: The SAM Database

When LSASS handles a local (non-domain) authentication attempt, it must validate the provided credentials against a trusted source. This source is the SAM (Security Account Manager) database, a file located at '\%systemroot\%system32\config\sam'. The SAM database serves as the definitive ledger for all local user accounts, storing their corresponding password hashes. Historically, this file was the primary target for login bypass techniques. To protect it during normal operation, the Windows kernel maintains an exclusive lock on the file, preventing any direct access or modification. This defence, however, is only active while Windows is running, which led to the prevalence of "offline attacks"—methods that involve accessing the SAM file from an external operating system to modify its contents.

C. The Physical Shield: BitLocker Full-Disk Encryption

The risk of offline assaults on the SAM database and other system files is primarily reduced through Full-Disk Encryption (FDE). BitLocker, Microsoft's native solution, offers full-disk encryption for the operating system partition, rendering all data, including the SAM database, cryptographically unreadable unless the appropriate decryption key is provided. With BitLocker enabled, an attacker who boots the machine from an external drive will find an inaccessible, encrypted partition. This technology effectively shifts the entire attack surface away from offline file system manipulation and onto the live, pre-boot, or running system, forcing an attacker to find a way to circumvent the authentication process while the system is active.

D. The Anchor of Trust: Hardware-Integrated Security

The security of BitLocker is ultimately dependent on the protection of its decryption key. To protect this vital secret, current Windows security is firmly rooted in the hardware of the system, creating a trust chain that starts as soon as the device is turned on.

1) *Trusted Platform Module (TPM):* This is a specialized, hardware-based security processor that serves as the foundational element of the trust chain. The TPM provides a secure, tamper-resistant environment for cryptographic operations. In a standard BitLocker configuration, the disk encryption key is "sealed" within the TPM.

2) *UEFI Secure Boot:* The TPM is designed to release the sealed key only if the boot process is verified as secure and unmodified. This verification is managed by UEFI Secure Boot, a firmware standard that ensures each component of the boot process (from the firmware itself to the OS bootloader) is cryptographically signed and trusted. If an attacker attempts to boot an unauthorized operating system (e.g., from a malicious USB drive), Secure Boot will block it. If they modify a signed component, the TPM's integrity checks will fail, and it will refuse to release the BitLocker key, thus preventing access.

3) *Virtualization-Based Security (VBS):* Beyond the boot process, hardware-based protection is extended to the live operating system. Modern iterations of Windows utilize Virtualization-Based Security (VBS) to protect essential components like LSASS. VBS leverages the CPU's virtualization capabilities to create a secure, isolated memory region where sensitive processes can operate, shielding them from threats and inspection even from a compromised kernel.

4) *Kernel DMA Protection:* To protect against advanced physical attacks that involve directly accessing system memory, Windows has introduced Kernel DMA (Direct Memory Access) Protection. This capability utilizes the system's IOMMU (Input-Output Memory Management Unit) to block unauthorized peripherals (for instance, those connected via high-speed ports like Thunderbolt) from gaining direct access to RAM. This mitigates a class of attacks that could otherwise be employed to extract encryption keys and other confidential information from memory.

This tightly integrated, hardware-anchored security model represents the foundation of modern Windows defence and is the primary reason why many traditional login bypass techniques are no longer effective.

III. A TAXONOMY OF WINDOWS LOGIN BYPASS TECHNIQUES

The multi-layered security architecture of Windows, as detailed in the previous section, has not eliminated login bypass techniques but has instead forced them to evolve in sophistication. Attackers have developed a range of methods designed to circumvent or exploit each specific defensive layer. These techniques vary from

straightforward offline file modifications to intricate hardware-dependent and protocol-based attacks. This section offers a classification of these approaches, organizing them according to their operational state and examining their mechanisms, consequences, and the tools that execute them.

A. Classification by System State

Bypass techniques can be categorized into two main types based on the state of the target system at the time of 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, complete, unrestricted and unfettered 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. These attacks directly target the stored credential data when it is at rest.

2) *Online and "Live" Attacks*: These attacks occur when the Windows OS is operational or in a pre-boot state where the hardware is 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 favoured in red team exercises as they maintain the existing condition of the system, including the information stored in RAM.

B. Analysis of Common Bypass Methods

The following is a detailed examination of specific 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 the Utility Manager) with the Command Prompt (cmd.exe). When the accessibility icon is activated on the login screen, a command prompt with SYSTEM-level permissions is opened in place of the genuine 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 regular standard, non-privileged user account to the one with administrative rights, or unlocking a disabled account. 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 to subvert software-level protections.

- *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. This technique directly targets the live processes we discussed in Section II, bypassing file system protections entirely.
- *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 an 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 introduced a new attack surface centred on its implementation and interaction with hardware.
   - *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 it exploits has been patched or mitigated by the architectural security improvements in modern Windows—a topic we will explore in the next section.

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

The failure of legacy bypass tools and the increasing complexity of modern attack methods are not coincidental. They are the direct results of a deliberate and fundamental architectural hardening of the Windows operating system. In recent years, Microsoft has shifted from a model of disparate security features to a deeply integrated, hardware-anchored defense-in-depth strategy. This section details the key defensive pillars of modern Windows 10 and 11, explaining precisely how they neutralize the attack vectors discussed in the previous section.

A. The End of an Era: The Nullification of Offline Attacks

For years, the most reliable and accessible bypass methods were offline attacks. As we saw in Section III, these techniques relied on booting a system from external media to gain direct access to the file system, thereby circumventing the protections of the running Windows kernel. The effectiveness of this entire class of attack has been systematically dismantled on properly configured modern systems by the synergistic combination of three technologies:

1) *UEFI Secure Boot:* This firmware-level standard serves as the first line of defence, creating a "chain of trust" by ensuring the device boots only using cryptographically signed and trusted software. This directly prevents an attacker from loading an unauthorized operating system from a USB drive to perform registry or SAM file manipulation.

2) *Trusted Platform Module (TPM):* The TPM provides a hardware root of trust that anchors the entire boot process. When utilized alongside BitLocker, the TPM safely holds the disk encryption key and is intended to only unlock it if the boot sequence is confirmed to be unchanged. Any attempt to tamper with the bootloader or circumvent Secure Boot will be detected, causing the TPM to withhold the decryption key.

3) *BitLocker Full-Disk Encryption (FDE):* This is the final and most decisive countermeasure to offline attacks. With the disk fully encrypted, an attacker who somehow manages to boot from an external OS is still unable to read or modify any files on the Windows volume. The data, including the SAM database and registry hives, is rendered cryptographically inaccessible.

Together, these three features form a near-impenetrable barrier against offline manipulation, effectively closing a major significant phase in the history of Windows login bypass and forcing adversaries to contend with the live system.

B. Fortifying the Live System Environment

Recognizing this tactical shift, Microsoft has invested heavily in hardening the live system, protecting critical authentication processes even when a user is not logged in.

1) *Virtualization-Based Security (VBS) and Credential Guard:* In response to in-memory attacks targeting LSASS, modern Windows employs VBS. This feature uses the CPU's hardware virtualization capabilities to create a secure, isolated region of memory. Credential Guard, a key component of this system, runs the LSASS process within this protected environment. This prevents even a

compromised kernel from directly accessing the memory of LSASS, thereby mitigating the credential-dumping attacks that were a staple of advanced adversaries.

2) *Kernel DMA Protection:* To counter sophisticated hardware attacks that use Direct Memory Access (DMA), modern systems implement Kernel DMA Protection. This feature leverages the system's IOMMU (Input-Output Memory Management Unit) to block unauthorized peripherals from gaining direct, unfettered access to system RAM. This is a direct countermeasure to the DMA-based attacks detailed in Section III, neutralizing a significant physical attack vector.

3) *Continuous Patching and a Dynamic Defence Posture:* The Windows ecosystem is a dynamic battlefield, with new vulnerabilities being discovered and patched continuously. The response to the Windows Hello biometric spoofing attack is a prime example. After researchers demonstrated that a rogue USB camera could bypass facial recognition, Microsoft issued a patch (addressing CVE-2021-34466) that introduced a "secure camera" protocol, ensuring the OS only trusts input from certified hardware. This ongoing cycle of vulnerability discovery and mitigation is a primary reason why specific bypass tools and exploits have a finite lifespan.

C. The Consequence: The Obsolescence of Legacy Tools

The culmination of these architectural improvements explains why the tools and techniques that once defined Windows login bypass are no longer effective.

1) Tools reliant on offline SAM modification (e.g., chntpw) are rendered inert by the cryptographic barrier of BitLocker.

2) Hardware exploits based on DMA attacks are blocked at the hardware level by Kernel DMA Protection.

3) The Windows Login Unlocker Pro PE is also ineffective on Windows 10 versions post-22H2 - is a direct illustration of this security evolution. Its failure strongly suggests that it exploited a specific software vulnerability within the live logon process. Such a flaw would have been recognized and neutralized by Microsoft through the cumulative security updates that define the modern Windows platform.

Therefore, the security of a contemporary Windows system is not derived from a single feature, but from a deeply integrated, multi-layered defence 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 architectural hardening detailed in the previous section represents a decisive victory for defenders against the traditional login bypass playbook. However, this has not ended the conflict; it has merely shifted the battlefield. The evolution from broad, simple attacks to highly specific and complex ones presents new challenges for offensive security professionals and simultaneously charts the course for future research. This section discusses the current state of this new conflict, the emerging attack vectors on the horizon, and the implications for defensive strategies.

A. The New Reality: The Challenge of the "Intact Bypass"

The central challenge for any attacker or red teamer today is achieving an "intact bypass" on a modern, fully-patched Windows 11 system. As demonstrated, the foundational offline methods are now largely neutralized by the combined defences of Secure Boot, TPM, and BitLocker. This forces attackers into the live environment, where they are met with formidable, hardware-anchored defences

like VBS-protected Credential Guard and Kernel DMA Protection.

This robust security posture means that a generic, universally effective login bypass tool is likely a relic of the past. The failure of older software like Windows Login Unlocker Pro PE on modern systems is a testament to this reality. A successful bypass is no longer a matter of using a known tool; it is predicated on discovering and exploiting zero-day or other specific, unpatched vulnerabilities within the complex interplay of software and hardware that constitutes the live authentication process. By their very nature, such exploits are transient and are quickly rendered ineffective by routine security patches.

B. Future Research and Emerging Attack Vectors

The academic literature and security research community point to several key areas where the next generation of bypass techniques are likely to emerge.

1) *Offensive Artificial Intelligence and Automated Exploitation*: The manual, human-driven process of vulnerability discovery and exploit chaining is slow and inefficient. The future of offensive security is likely to be driven by artificial intelligence. Research has already demonstrated the feasibility of training a deep reinforcement learning agent to autonomously perform 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 far more rapidly and adaptively than a static script or even a human operator. This signals a paradigm shift from manually executed techniques to autonomous agents capable of performing red team operations at machine speed.

2) *The Biometric Attack Surface*: While a significant improvement over passwords, Windows Hello has introduced a new and inherently personal attack surface. As demonstrated by the "Windows Hell No" vulnerability and research into biometric spoofing via rogue USB devices, the implementation of biometric systems is fraught with potential weaknesses. While Microsoft has patched specific flaws, future research will likely focus on:
    1. Exploring side-channel attacks against biometric sensors to leak data.
    2. Exploiting vulnerabilities in the biometric enrolment and database management processes.
    3. Developing novel methods to inject spoofed biometric data that bypass hardware-level trust and integrity checks.

3) *Protocol-Level Weaknesses*: Even when the individual OS components are secure, the complex authentication protocols that bind them together can contain design flaws. The 2015 research that demonstrated a bypass of BitLocker did not attack the encryption itself, but rather a subtle weakness in the Kerberos password reset protocol that allowed an attacker to poison the cached credentials of a domain-joined machine. This serves as a powerful reminder 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 entirely.

C. Implications for Defensive Security

This survey of evolving bypass techniques provides a critical roadmap for defensive "blue teams" and system administrators. The key takeaway is that security is not a static state but a continuous process of adaptation.

1) *Configuration is Key:* The effectiveness of the advanced defences in modern Windows is entirely contingent upon their proper deployment. Administrators must ensure that Secure Boot, TPM, BitLocker, and Credential Guard are enabled and correctly configured across their enterprise fleet to realize their full protective potential.

2) *The Importance of Detection and Auditing:* Since a successful modern bypass likely indicates a

specific, unpatched vulnerability, robust system auditing is more critical than ever. The Windows Registry and Event Logs contain numerous digital artifacts useful for forensic analysis that can indicate a failed or successful attack. Monitoring for unusual registry modifications, anomalous login events, or unexpected hardware enumeration can provide the early warnings necessary to detect a sophisticated intrusion.

3) *Preparing for the AI Arms Race:* The emergence of offensive AI necessitates the development of AI-driven defensive systems. Several researchers have proposed that future security solutions must 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 required to counter autonomous attackers.

In summary, the field of Windows login bypass is far from resolved. While the barrier to entry has been raised considerably, new frontiers in AI, biometrics, and protocol analysis are continuously emerging, ensuring that the strategic contest between aggressors (attackers) and protectors (defenders) will persist for the foreseeable future.

VI. Conclusions

This survey has chronicled the strategic evolution of Windows login bypass techniques, examining the persistent interplay between offensive methodologies and defensive innovations. Our investigation began by documenting the foundational era of "destructive" offline attacks, which for years provided a reliable means of compromising systems by directly targeting the Security Account Manager (SAM) database and the Windows Registry.

However, the central and overriding conclusion of this paper is that the security landscape has been fundamentally reshaped. The architectural hardening of modern Windows operating systems—achieved through a deeply integrated, hardware-anchored defense strategy combining UEFI Secure Boot, the Trusted Platform Module (TPM), BitLocker, and Virtualization-Based Security (VBS)—has rendered this entire class of traditional offline attacks largely ineffective on properly configured systems.

This defensive consolidation has forced a clear and decisive migration of the attack surface. The focus of sophisticated adversaries has shifted away from the offline file system and onto the live, running system. As this survey has detailed, modern bypass techniques are now predicated on discovering and exploiting specific, often transient, vulnerabilities within the complex components of the active authentication environment, such as the Windows Hello biometric framework, associated hardware interfaces, and the underlying network authentication protocols.

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 simple or probable proposition. Specialized tools that once promised such capabilities are inevitably built upon specific, patchable flaws, explaining their limited lifespan and obsolescence against newer OS versions. The future of this domain does not lie in a single "silver bullet" exploit, but in the emerging fields of automation and artificial intelligence, where autonomous agents will be developed to discover and chain vulnerabilities in real-time. This marks the start of a new, more intricate phase in the continuing security competition between aggressors (attackers) and protectors (defenders) in the Windows environment..

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/.