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

Atharva Jagtap[1], Prachi Dalvi[2]

[#]*Computer Science and Engineering, Fr. Conceicao Rodrigues College of Engineering*
Bandra, Mumbai, India

[1]atharvaj365@gmail.com

[3]prachidalvi@gmail.com

Abstract— **User authentication is the foundation of modern digital security in Microsoft Windows. For people working in penetration testing, red teaming, or digital forensics, breaking through this barrier is usually the most difficult task. This paper describes the main tools and techniques that attackers use to circumvent the Windows login. We trace the development of these attacks from classic attacks on the Security Account Manager (SAM) to more advanced, "live" tactics that target hardware interfaces or exploit vulnerabilities in authentication systems such as Windows Hello. The situation improved with Windows 10 and 11. The stringent, hardware-backed security of many of the older methods-think Secure Boot, Trusted Platform Module, and Virtualization-Based Security has made them obsolete. Attackers now focus on vulnerabilities in components of the active system. For people working in penetration testing, red teaming, or digital forensics, breaking through this barrier is usually the most difficult task. This paper describes the main tools and techniques that attackers use to circumvent the Windows login. We trace the development of these attacks from classic attacks on the Security Account Manager (SAM) to more advanced, "live" tactics that target hardware interfaces or exploit vulnerabilities in authentication systems such as Windows Hello. Windows 10 and 11 improved the situation. The stringent, hardware-backed security of many of the older methods-think Secure Boot, Trusted Platform Module, and Virtualization-Based Security has made them obsolete.**

Keywords— **Windows Security, Login Bypass, Red Teaming, Penetration Testing, SAM Database, Windows Hello, Credential Theft**

I. INTRODUCTION

A. The Evolving Battlefield of Windows Authentication

Microsoft Windows is the cornerstone of both personal and professional computing. The first line of defense against unauthorized access to your data is user authentication. At first, Windows used straightforward passwords. Things were different with Windows 10 and 11. More security layers are now added with multi-factor authentication. Windows goes beyond that. In order to eventually do away with the need for passwords, Microsoft is developing features like Windows Hello. This system incorporates biometric authentication which, when combined with hardware-based security features such as the Trusted Platform Module (TPM), offers enhanced security to counter threats that are evolving and increasingly sophisticated.

B. Problem Statement and Motivation

The systems have changed from earlier versions' password-based approaches to Windows 10 and 11's advanced multi-factor authentication features. However, the main goal for digital forensic analysts and offensive security specialists is to gain access to an operational system while it is still in use without changing its state. Important data, such as session tokens, cryptographic keys, and running processes, are kept in a live computer's volatile memory (RAM), which is permanently deleted when the system is turned off or shut down. The techniques employed by the traditional login bypass programs are directly at odds with this necessity.

Traditionally, recovery suites like Hiren's BootCD, have been using "destructive" techniques like modifying the Security Account Manager (SAM) database. In order to directly modify the SAM database offline using this method, it requires booting from external media, it allows to reset, blank, create or change user passwords. Such a modification is often forbidden in red team engagements or forensic examinations because it alters evidence, alerts defenders of a hack, and may render user-keyed encrypted material unavailable.

This situation has led to the necessity for intact and non-destructive bypass approaches that permit access to a system while maintaining the original user credentials. Specialized bypass tools like "Windows Login Unlocker Pro PE," which claimed to provide this capability, indicated that such methods were possible, but the tool reported

inefficiency against the most recent versions of Windows (Windows 10 post-22H2). however, the tool reported an inefficiency against the most recent versions of Windows (Windows 10 post-22H2) which highlights a significant capabilities gap, and that is what drives our investigation to document the most recent advancements in Windows login bypass techniques and to examine the reasons why older methods won't work against today's security hardened systems.

II. FOUNDATIONAL WINDOWS SECURITY ARCHITECTURE

One must first comprehend how a great login functions in order to comprehend how a Windows login could be compromised. Entering a password, PIN, or using a biometric scanner initiates a complex, multi-step verification process. Several key elements oversee this process. This procedure is safeguarded by underlying hardware level security and includes both live in-memory validation and checks against stored credential data. By examining its four main architectural pillars, this section will break down that process.

We will begin with the Local Security Authority Subsystem Service (LSASS), which is the main process in charge of handling live authentication attempts. The historical ledger used to verify local credentials, the Security Account Manager (SAM) database, will Entering a password, PIN, or using a biometric scanner initiates a complex, multi-step verification process. Several key elements oversee this process. This procedure is safeguarded by underlying hardware and physical level security and includes live, in-memory validation in addition to checks against stored credential data. This section will break down that process by examining its four basic architectural foundations.

Since the layers are made up of LSASS, SAM, BitLocker, and TPM, they are inevitably at the top of every attacker's list of targets. The purpose and internal operations of each component are explained in the ensuing subsections, which set the stage for the bypass strategies covered later in this essay.

A. The Core Orchestrator: LSASS and the Logon Process

The Windows OS's live user authentication is managed by the Local Security Authority Subsystem Service (LSASS), which appears as the process lsass.exe. The LSASS logs the information for the verification when a user inputs their login credentials into the Logon User Interface (LogonUI.exe). After then, it uses a number of authentication packages to confirm the user's identity. In order to verify the credentials, LSASS usually establishes a conversation with the domain controller on a network-connected system by tapping the Kerberos protocol. The responsibility is transferred to an authentication package on a standalone locally operated machine. As LSASS maintains data hashes in memory, it is a prime target for sophisticated attacks like memory dumping, which attempt to retrieve login credentials while the system is operating.

B. The Credential Store: The SAM Database

The SAM (Security Account Manager) database, hidden in ‘%systemroot%/system32/config/sam`, is a trusted store that LSASS must check the credentials against when it gets a non-domain login attempt. For each user account, the SAM serves as a ledger while maintaining the associated password hashes. This file has been the focus of methods that try to get around the login procedure over the years. By keeping a lock on the file while it is running, the Windows kernel successfully stops any direct read or write attempts. The frequency of "offline attacks," which entail launching a separate operating system and getting access to the SAM file in order to modify it, can be explained by the fact that this security is only accessible when Windows is active.

C. The Physical Shield: BitLocker Full-Disk Encryption

Mitigating assaults, on the SAM database and other system files largely depends on full-disk encryption (FDE). Microsoft’s native BitLocker encrypts the operating-system partition turning

every piece of data-including the SAM-into gibberish unless the correct decryption key is supplied. With BitLocker active a would-be intruder who tries to boot the machine from a drive will simply encounter a encrypted partition. This approach forces an attacker to find a means to get around authentication while the system is operating by shifting the entire attack surface from offline file-system tinkering to the pre-boot or running machine.

D. The Anchor of Trust: Hardware-Integrated Security

The decryption key's protection ultimately determines BitLocker's security. In order to secure this vital secret, current Windows security is firmly embedded 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):* Think of it as a hardware guardian, the anchor of the trust chain. The TPM is hardware-based security processor that serves as the foundational element of the trust chain. It creates a locked-down, tamper-proof enclave for work. In a BitLocker setup the disk-encryption key is bound 'sealed' to the TPM.

2) *UEFI Secure Boot:* The TPM is designed such that it will only hand over its sealed key once it confirms the boot chain is intact and unaltered. This check is performed by UEFI Secure Boot, a firmware-level mechanism that insists every component of the startup process—, from the firmware itself, to the OS bootloader—be cryptographically signed and trusted. If an attacker tries to launch an operating system for example by plugging in an USB stick Secure Boot will intervene and block it. Should a signed component be tampered with the TPM's integrity checks will flag the alteration. Refuse to hand over the BitLocker key effectively denying access.

3) *Virtualization-Based Security (VBS):* Beyond the boot process the hardware-based shield doesn't just disappear-it carries on protecting the operating system. Contemporary Windows releases employ VBS to wrap pieces such, as LSASS, in a cocoon. VBS creates a memory enclave where sensitive workloads can operate by utilizing the CPU's virtualisation capabilities, protecting them against intrusions and even a hacked kernel.

4) *Kernel DMA Protection:* Windows has implemented Kernel DMA (Direct Memory Access) Protection to guard against sophisticated physical assaults that involve directly accessing system memory. This feature prevents unauthorized peripherals (such as those connected via high-speed connectors like Thunderbolt) from directly accessing RAM by using the system's IOMMU (Input-Output Memory Management Unit). By doing this, a class of attacks that may otherwise be used to retrieve encryption keys and other private data from memory are lessened.

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

III. A TAXONOMY OF WINDOWS LOGIN BYPASS TECHNIQUES

As explained in the preceding section, Windows' multi-layered security design has encouraged login bypass techniques to become more sophisticated rather than eliminating them. Attackers have created a variety of techniques to get around or take advantage of each distinct protective layer. These methods range from simple offline file changes to complex hardware-dependent and protocol-based assaults. This section provides a categorization of various methods, grouping them based on their operational state and analyzing their mechanisms, outcomes, and execution instruments.

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 favored 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 centered 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)*: Specialized commercial tools are said to provide more sophisticated, non-destructive bypass capabilities than standard recovery suites. "Windows Login Unlocker Pro PE," which you discovered through your inquiry, claims to install a bypass mechanism that permits logging into an account without typing a password at all while maintaining the original password. According to your research, this particular utility is only useful with previous versions of Windows (before Windows 10 22H2). This strongly implies that the architectural security enhancements in contemporary Windows have addressed or reduced the underlying weakness it exploits—

a subject we shall discuss in the following section.

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

Legacy bypass tools aren't just failing by chance. Modern attacks haven't suddenly grown more complex on their own. Microsoft has completely overhauled the core of Windows. It's not just about adding extra security features but the defense that runs all the way down to the hardware level. In this section, we will unpack the main security pillars in Windows 10 and 11 and show how they block the attacks we discussed earlier.

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

For a long time, offline attacks were the go-to option for bypassing security. An attacker could just boot up the system from some external drive, get straight into the file system, and sidestep all the protections of the running Windows kernel. But things have changed. On modern systems that are set up right, a trio of technologies work together to pretty much shut these attacks down:

1) *UEFI Secure Boot:* It serves as the device's gatekeeper. It ensures that nothing dubious gets through and programs only with authorized cryptographic signatures can load at boot. It prevents attackers from simply inserting a USB device and launching an unauthorized operating system and tamper the registry or SAM files of the OS.

2) *Trusted Platform Module (TPM):* This module anchors the entire boot process with a hardware root of trust. The TPM securely stores the disk encryption key when used in conjunction with BitLocker and is designed to unlock it only upon verification that the boot sequence remains unaltered. The TPM will withhold the decryption key if it detects any attempt to meddle with the bootloader or get around Secure Boot.

3) *BitLocker Full-Disk Encryption (FDE):* The last and most effective defense against offline attacks is BitLocker Full- Disk Encryption (FDE). Even if the disk is completely encrypted, any files on the Windows volume cannot be read or altered by an attacker who manages to boot from an external operating system. The information is presented, including registry hives and the SAM database.

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

Microsoft has made significant investments to secure the live system in recognition of this tactical shift, safeguarding crucial authentication procedures even when a user is not signed in.

1) *Virtualization-Based Security (VBS) and Credential Guard:* VBS stops in-memory attacks against LSASS in modern Windows. This feature isolates and secures a section of memory by utilizing the CPU's hardware virtualisation capabilities. Credential Guard, an essential component of this system, performs the LSASS process in this secure environment. By preventing even a damaged kernel from directly accessing the memory of LSASS, this lessens the impact of credential-dumping attacks, which were a common strategy employed by skilled adversaries.

2) *Kernel DMA Protection:* Modern systems use Kernel DMA Protection to protect against sophisticated hardware attacks that use Direct Memory Access (DMA). This feature prevents unauthorized devices from having direct, unrestricted access to system RAM

by using the system's IOMMU (Input-Output Memory Management Unit). This removes a major physical attack vector and directly counters the DMA-based attacks described in Section III.

3) *Continuous Patching and a Dynamic Defense Posture:* The Windows environment is a dynamic battlefield because new vulnerabilities are continuously discovered and fixed. The response to the Windows Hello biometric spoofing attack is among the better examples. After researchers demonstrated that a rogue USB camera could evade facial recognition, Microsoft released a patch (fixing CVE 2021-34466) that added a "secure camera" protocol, ensuring the operating system only takes input from authorized hardware. This ongoing cycle of vulnerability discovery and mitigation is one of the primary reasons why some bypass tools and exploits have a short lifespan.

C. The Consequence: The Obsolescence of Legacy Tools

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

1) Tools that rely on offline SAM modification (like chntpw) are rendered inactive by BitLocker's cryptographic barrier.

2) At the hardware level, Kernel DMA Protection stops DMA-based hardware vulnerabilities.

3) The fact that Windows Login Unlocker Pro PE is inoperable on Windows 10 versions after 22H2 is another glaring illustration of this security progress. It is very likely that it exploited a specific software vulnerability in the live logon process given its failure. Such a vulnerability would have been found and corrected by Microsoft's cumulative security updates, which are a feature of the current Windows platform.

A generic, one-size-fits-all "live login bypass" is therefore a far more difficult and frequently impossible undertaking since the security of a modern Windows system is derived from a thoroughly integrated, multi-layered defense rather than from a single feature.

V. DISCUSSION: CHALLENGES AND FUTURE RESEARCH DIRECTIONS

The architectural hardening described in the previous section is a big win for defenders against the old login bypass playbook. But this hasn't ended the war; it has just moved the battlefield. The shift from broad, simple attacks to very specific and complicated ones makes things harder for offensive security professionals and also sets the stage for future research. The present status of this novel conflict, new attack methods, and the consequences for defensive strategies are covered in this section.

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

Finding a "intact bypass" on a modern, fully patched Windows 11 system is currently the biggest challenge facing any attacker or red teamer. As shown, the traditional offline methods are now essentially useless due to the combined security features of Secure Boot, TPM, and BitLocker. Because of this, attackers are forced to enter the live environment, where they have to deal with strong hardware-based defenses like Credential Guard and Kernel DMA Protection.

This strong security posture makes it unlikely that a generic, universally effective login bypass tool is still around. This is true because older software like Windows Login Unlocker Pro PE doesn't work on newer systems. A successful bypass is no longer just about using a known tool. It now depends on finding and taking advantage of zero-day or other specific, unpatched vulnerabilities in the complicated interaction between software and hardware that makes up the live authentication process. Because of their nature, these kinds of exploits are only temporary and are quickly made useless by regular security patches.

B. Future Research and Emerging Attack Vectors

The security research community and scholarly literature identify a number of crucial areas where the next wave of bypass techniques is probably going to appear.

1) *Offensive Artificial Intelligence and Automated Exploitation*: Finding vulnerabilities and chaining exploits by hand is a time-consuming and inefficient process. In the future, offensive security is probably going to be fuelled by artificial intelligence. It has already been demonstrated that it is possible to train a deep reinforcement learning agent to perform local privilege escalation on its own. Such an agent can learn to identify system misconfigurations (like unquoted service paths or hijackable DLLs) and execute the optimal course of action far more rapidly and adaptively than a static script or even a human operator. With the ability to perform red team tasks at machine speed, self-governing agents have replaced manual methods, marking a paradigm shift.

2) *The Biometric Attack Surface*: Despite being a significant improvement over passwords, Windows Hello has introduced a new and inherently personal attack surface. Studies on biometric spoofing through rogue USB devices and the "Windows Hell No" vulnerability demonstrate the numerous potential vulnerabilities associated with the use of biometric systems. Despite the fact that Microsoft has resolved some problems, future studies 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 if the OS components are secure, there may be design flaws in the complex authentication protocols that link them. The 2015 study demonstrated how to get around BitLocker without attacking the encryption itself by taking advantage of a small vulnerability in the Kerberos password reset protocol that allowed an attacker to tamper with the cached credentials of a domain-joined machine. This serves as a powerful reminder that security research will continue to concentrate on complex protocols like NTLM and Kerberos because of the potential for new attacks that totally circumvent OS-level defenses.

C. Implications for Defensive Security

For defensive "blue teams" and system administrators, this survey of developing bypass techniques offers a vital road map. The main lesson is that security is an ongoing process of adaptation rather than a fixed state.

1) *Configuration is Key:* The effectiveness of Windows' advanced defenses depends on how they are used. Administrators must ensure that Secure Boot, TPM, BitLocker, and Credential Guard are activated and configured correctly in order to completely secure their enterprise fleet.

2) *The Importance of Detection and Auditing:* Because a successful modern bypass most likely indicates a specific, unpatched vulnerability, thorough system auditing is more crucial than ever. The Windows Registry and Event Logs contain numerous digital artifacts that are useful for forensic analysis and can show whether an attack was successful or unsuccessful. By monitoring for unusual hardware enumeration, unusual login behavior, or unusual registry changes, one can find the early warning indicators needed 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 believe that in order to analyze system and command behavior in real time, machine learning must be incorporated into future security solutions. By analyzing a PowerShell command sequence before it is executed to determine whether it matches a known attack pattern, for example, an AI-powered defense could proactively block the threat. This suggests that in an inevitable "arms race," autonomous defensive agents will be required to counter autonomous attackers.

In conclusion, the field of Windows login bypass still needs a lot of work. Despite the considerable rise in entry barriers brought about by the continuous development of new frontiers in AI, biometrics, and protocol analysis, the strategic conflict between aggressors (attackers) and protectors (defenders) will persist for the foreseeable future.

VI. Conclusions

The strategic development of Windows login bypass methods is documented in this survey, which also looks at the ongoing interaction between offensive and defensive innovations. Our research started by recording the earliest "destructive" offline attacks, which directly targeted the Windows Registry and the Security Account Manager (SAM) database and were a dependable way to compromise a system for many years.

Nonetheless, this paper's main and most important finding is that the security environment has undergone a significant transformation. This entire class of traditional offline attacks is now largely ineffective on properly configured systems due to the architectural hardening of modern Windows operating systems, which is accomplished through a deeply integrated, hardware-anchored defense strategy that combines BitLocker, the Trusted Platform Module (TPM), UEFI Secure Boot, and Virtualization-Based Security (VBS).

This defensive consolidation has resulted in a clear and decisive migration of the attack surface. Instead of focussing on the offline file system, advanced adversaries are now focussing on the live, functioning system. According to this survey, current bypass techniques now depend on locating and exploiting specific, often transient vulnerabilities in the complex components of the active authentication environment, such as the Windows Hello biometric framework, associated hardware interfaces, and the underlying network authentication protocols.

As a result, this paper validates the original research hypothesis: it is no longer simple or likely to find a universal, non-destructive ("intact") login bypass tool for a fully-patched, contemporary Windows 11 system. A limited lifespan and obsolescence against newer OS versions can be explained by the fact that specialized tools that once promised such capabilities are invariably built upon specific, patchable flaws. The development of autonomous agents that can identify and chain vulnerabilities in real-time is what will shape this domain's future, not a single "silver bullet" exploit. The ongoing security competition between aggressors (attackers) and protectors (defenders) in the Windows environment is about to enter a new, more complex phase.

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 MITER 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 Center (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/.