Addressing the Inadequacies of Role Based Access Control (RBAC) Models for Highly Privileged Administrators: Introducing the SNAP Principle for Mitigating Privileged Account Breaches

Samuel Moses, Dale C. Rowe, Sarah A. Cunha
Brigham Young University Cyber Security Research Laboratory

Abstract

In this paper, we discuss how RBAC systems fail to protect highly privileged accounts used for system administration. We introduce how Secondary Non-Admin Privileged (SNAP) accounts can mitigate a variety of attacks targeting privileged accounts. Both justification and a methodology for implementing this approach are presented along with case-studies showing how real attacks can be mitigated. Three different variations we have termed i-SNAP, s-SNAP and t-SNAP are compared. Other studies have shown that over 92% of critical vulnerabilities require administrative access and we present multiple case-studies that demonstrate the effectiveness of our proposed solution. Also discussed are procedural, technical and educational processes that will increase the effectiveness of this approach. We conclude with a critical assessment of the SNAP approach and include its potential limitations.

1. Introduction

System administrators are an essential asset to the organizations they serve. In the same way a steam-train engineer ensures the smooth running of a locomotive, system administrators ensure that computer systems are optimized for the tasks they must perform. The responsibilities of a systems administrator may include systems engineering, networking and security; in addition to implementing, maintaining and monitoring an organizations infrastructure [1]. They are among the most trusted of all system users; inherently having privileged access to nearly all organizational IT infrastructure.

When system administrators have privileged access throughout a company, it is extremely important for them to protect their login credentials. Current best practices recommend that a personal set of administrative credentials are used rather than shared accounts. This enforces accountability and can enable non-repudiation. These accounts are often provided by a Single-Sign-On (SSO) provider and grant access to a multitude of system types. To attackers however, system administrator user accounts represent the 'keys to the kingdom', and can lead to the total exposure of an organization's intellectual property, systems and customer information.

Avecto in 2013 conducted a vulnerability study on Microsoft to test how risk could be mitigated by limiting user privileges. Users whose access was configured to have fewer user rights on the system were less impacted by vulnerabilities than users who operated with administrative user access. It was found that out of all the vulnerabilities reported, 60% were mitigated simply by removing administrator rights. However, Microsoft ranks vulnerabilities according to their potential impact in a system breach. When Avecto narrowed the study to focus on the most serious vulnerabilities, identified by Microsoft as critical, (this severity consisting of approximately half of the total vulnerabilities in the study) 92% were mitigated by the user not having administrator access [2]. This demonstrates the need to guard system administrator accounts by using risk mitigation techniques to protect their credentials.

Traditional role-based-access-control or RBAC systems use the role of a user to determine their access to objects. The limits user access within their job responsibilities and helps prevent unauthorized access to unnecessary objects. While generally perceived as an effective security mechanism for users, we shall demonstrate how the limitations and flaws of RBAC can lead to complete system compromise.

In addition to demonstrating the failings of RBAC as an access control model for highly privileged accounts, we shall show a simple approach to extending RBAC to provide significantly enhanced security:

A Secondary Non-Admin Privileged (or SNAP) account is a technique that can be used to mitigate the risk of compromised system administrator credentials. It adds an additional layer of protection for system administrators as they perform many of their tasks. This process can provide a logical separation of duties for administrators with privileged access. In particular, we have found an increased benefit to administrators who are involved more with day-to-day helpdesk problems.

In this paper, we will explain the value of using a SNAP account and its primary use cases. For this it is important to understand the role of a system administrator, the credentials typically required for their job duties, and why it is so important to protect them. We will then review today's best practice recommendations and examine how SNAP accounts can provide a methodology to increase practical security. Finally, we will review a case study of a system administrator who had their administrative credentials stolen and demonstrate how the proper use of SNAP accounts could have prevented a major systems breach.

2. Role Based Access Control (RBAC)

Role-based-access-control arose due to the need of a more flexible method of access control than was previously in use. While discretionary and mandatory access-control mechanisms offered a simple way of enforcing access-control based on the classification or ownership of information objects they struggled to gain acceptance outside of military and research systems. Limitations in scalability and flexibility meant a new approach was required [3]. This led to the proposal of a new approach to access control based on user roles that was standardized by the National Institute for Science and Technology (NIST) as 'Role Based Access Control' or RBAC [4].

RBAC offers several models to provide access-control based on user roles. The most common approach to its implementation involves identifying required privileges for object access and defining these within user roles. Done properly, RBAC offers excellent scalability, flexibility and accountability.

Under an RBAC model, user accounts are added to groups that are defined at a departmental or organizational level. A group membership is defined as a role and allows the user to inherit access to the objects necessary to perform their job function. For example, a developer may be granted write-access to a code repository and read-access to their team's schedule and resource management software. While a project manager would require the ability to modify the team schedule and resource allocations, they would not typically need to submit code-changes. They may however need to access timesheets and submit modifications for their team's hours which in turn may be read by an accountant running the organizations payroll.

The RBAC model offers a simple way to offer all the above users required access to perform their day-to-day job functions that provides both confidentiality and integrity. Some extensions to RBAC such as TRBAC have been proposed to offer time-based access (for example, limiting the accounts access to payroll until the payroll needs to be run) further increasing security [5].

While appropriate for the above use cases, current operational risks in cybersecurity lead us to challenge the suitability of traditional RBAC systems when it comes to highly-privileged access. As we shall demonstrate, the role of a systems administrator by definition undermines security in RBAC systems to a severe extent. Not only is compromise possible, it is in many instances likely to lead to a total loss of confidentiality, integrity and even availability depending on an attackers motives. In these breaches, it is even possible to destroy forensic artifacts that may be necessary in the incident-response process.

3. The System Administrator Role

System administrators are the workers behind the scenes responsible for keeping services up and running. They monitor, maintain, and engineer a company's information technology infrastructure, networking, and security. Their tasks are varied depending on an organization's model; they often work in diverse elements of infrastructure to maintain the health of their systems.

The roles of a system administrator may also include installing hardware such as routers, switches and firewalls; management of server and client software such as directory services, desktop applications and databases; performing maintenance including patches, updates and repairs. They are responsible for systems integration and making sure the entire network and computer system are operating together as they should. It is their duty to collect data in order to evaluate the network or system's performance to improve upon the systems, making them more reliable, secure and responsive.

Typically systems administrators manage federated user databases such as Active Directory or OpenLDAP and have the ability to add and remove user and service accounts. They manage organizational RBAC systems by granting access-control privileges to control access to objects such as information, networks and computer systems. They also train users on the proper use of these systems [6]. In some companies, system administrators are tasked with the day-to-day helpdesk problems that plague users, such as login issues, printer errors and end-user desktop support.

The role of a systems administrator thus requires elevated permissions. While they might not have direct access to login to an accounting software product, they will almost always have access to the underlying filestore, encryption keys, workstations, servers and databases used by the software. Often they will have the ability to grant access to the software to new users. If an administrator account possesses has 'effective' access in this way, the ability to directly

login to the software becomes insignificant. Loss of administrative credentials can lead to a complete accounting breach.

This is but one simple example of the level of access systems administrator accounts have to access a variety of computer access terminals and network devices spread throughout an organization.

Administrative roles must possess greater access privileges on a machine then typical user roles. Typical user accounts can make limited changes to their own account (such as resetting their own password), but cannot enact changes to other accounts. An administrator account has certain access privileges to effect changes on other user accounts. These accounts can also change security settings, install software and hardware, access all files on any computer under their jurisdiction and make changes to other user accounts [7].

In an effort to reduce risk, some companies separate the roles of the work between different administrative roles. For example, Google Apps comes preset with admin roles, each with their own defined roles and tasks they can accomplish. A Groups Admin has full control over Google Groups created in the admin console; a User Management Admin can perform all actions on users who aren't administrators and, optionally, a few other administrative roles that can be assigned. However, in all of these cases there is always at least one Super Admin. Super Admin has access to all features in the Admin console and can manage every aspect of an organization's accounts [8].

While there is a clear need for these system administrator roles, it is important to protect these accounts. When the wrong person, such as a malicious attacker, is able to gain access to administrative accounts, the results are frequently devastating. It is unsurprising that privileged accounts are typically the primary target of malicious attackers. With these privileges a malicious attacker has the ability to view, create, install, or run any software that they want. An attacker could even set up backdoors to slowly siphon off information while purging logs of the event, remaining hidden from systems administrators and auditors. With such access, an attacker may create additional administrative accounts for themselves that could go unnoticed for months or even years.

In summary, system administrators require privileged access to fulfill their role within a company. Because of this, they are a prime target for malicious attackers. Effective strategies and training are critical to protect their credentials and their companies.

4. Best Practices in Use Today

The value of administrative accounts has been known for some time, however there are limited best practices available today to protect these accounts. These best practices can mitigate the threat of the credentials being stolen. The following is a summary of current best practices to help secure and protect administrative accounts from compromise.

4.1. Limit Super Users (Delegation)

One of the best ways to protect an administrator's access on a system is to limit who can have it. The fewer privileged accounts, the smaller the attack surface and thus the smaller the risk. The principle of delegation involves a minimal set of system administrators with full system administrative privileges while maintaining limited scope privileges for others. A delegated administrator may have access to resources within their jurisdiction but limited access to any other systems.

4.2. Physical Security

SANS Institute recommends that system administrators should "make sure to install systems in a secure location where only authorized personnel are allowed. If there is physical access to the system console or the computer, it is easy for anyone to break-in or misuse [it]" [9]. This simple practice should be strictly followed. If a malicious attacker can get physical access to a host then they can gain complete control over it. Risk factors go from relatively benign to complete information breaches such as turning off, unplugging, equipment destruction, or hardware and/or information theft to analyze the data at a later date. Physical access can lead to using known procedures or back door entry processes to break into the system using the console. While these procedures allow legitimate system administrators to recover from forgotten credentials or accidental lockouts, it also underlines the importance of making sure servers and equipment are in a physically secure location such as a secure network operations center or a server room.

In the same way, "System Administrator's terminals or the terminals used by administrators are of high risk if they are not secured" [9]. If an attacker get access to system administrator's terminal, they will likely have access or be able to get access to multiple systems. Additionally, administrator terminals generally keep multiple sessions or windows to different systems open to carry out administrative tasks. Thus an attacker getting access to such a workstation may gain access to many others remotely. Therefore, system administrator's workstations must also be located in a secured area.

4.3. Securing Unused Workstations

Current best practices state that all users should lock their workstation when leaving it for any length of time. For system administrators this is even more critical. Locking a computer helps to protect personal

data from being stolen and an account from being impersonated or improperly accessed. As with the other attacks discussed, a system administrator's account provides access to multiple people's personal data and a company's intellectual property and customer information. Malicious attackers know this and know to target system administrators. This ability to spread through administrative systems can be impeded by adopting a locked screen policy. Technical controls such as screen timeouts can help by automatically locking a screen after a period of time has passed without user activity. However, manual screen locking should be habitualized by those with privileged access to minimize exposure.

While perseverance and diligence in forming secure habits can go a long way, an attacker may only need one mistake to gain system access. This is why using good habits should be supplemented by technical controls such as screen timeouts to lock screens.

4.4. Passwords

"System Administrators should be very cautious about root or administrator passwords" [9]. Today, passwords are the most widely used form of authentication despite offering relatively poor security as a single authentication factor. In a study of 6 million accounts a mere 10,000 common passwords gained access to 99.8% of accounts [10]. It is becoming easier and easier to crack passwords as computers and graphics cards increase in power.

The need for strong passwords to defend against these attacks has become more important as the ability of computers to brute force password hashes has increased. A password policy, especially one set up with technical controls to meet certain requirements, can help mitigate these attacks. A strong password may include:

- At least three of the four following character classes:
    - Lower case characters
    - Upper case characters
    - Numbers
    - "Special" characters (e.g. @#$%^&*()_+|~-=\`{}[]:";'<>/ etc…)
- At least fifteen alphanumeric characters.

There are more rules that should be followed when dealing with passwords:

- Set up password recovery options and keep them up to date. Ensure that hashing is used to protect recovery question answers.
- Never share passwords with others.
- Never reuse passwords between systems.
- Never write down passwords.
- Avoid password reuse i.e. have a different password for domain administrators and local administrators.
- Always remember to log off, especially on public workstations.
- Use up-to-date security software to prevent malware.
- Avoid entering sensitive passwords on untrusted systems.
- Change passwords periodically.

User education in password creation can help users to create passphrases that are both secure and memorable, thus reducing the risk of users writing down or reusing passwords.

4.5. Beware of Social Engineering

System Administrators need to be extremely vigilant against any attempt to gain access via social engineering. Due to their high privileges, they are prime targets for this form of attack. Social engineering is the use of persuasive techniques that target human nature to allow an attacker to obtain and learn information about the system (passwords, server names, version of antivirus, etc.).

While there is no way to be fully protected from a social engineering attack, a company can put in place policies, procedures, and controls to help manage the risk: Locking screens, keeping workstations and terminals in a secure area, and limiting physical access can aid with mitigating this risk. However social engineering allows more attack vectors as attacks can target regular users who may be used in turn to target system administrators.

Phishing is a method of social engineering in which an attacker uses a legitimate looking, but fake, email or website to lure people in an attempt to gather personal information and credentials. Since people often reuse their credentials, hackers can often hack a non-essential website or create a spoofed website and attempt to reuse these stolen credentials in attacking other critical accounts. In a study that analyzed online banking passwords, it was found that 73% of users shared their online banking password with at least one non-financial website [10]. When passwords are reused an otherwise secure system may be compromised by a password capture outside of that scope. Passwords should not be reused, and sites used by system administrators ought to be verified as secure.

Using policies and technical controls effectively can provide a strong defense against social engineering. Security policies can be developed and established so that employees have a clear understanding of what is expected of them. This in turn requires educating and training staff to understand the importance of security and policies. Having technical

controls to help enforce these policies is the best practice whenever possible. An example of this could consist of educating users to lock their screens while at the same time implementing technical controls that lock the screen after a period of inactivity.

Staying up to date with security and keeping policies current is paramount in developing and maintaining a secure environment. User training and education on current trends in social engineering and performing regular penetration tests that include this attack vector will help mitigate the risk of a social engineering attack.

4.6. Awareness Training

One of the greatest risks to an organizations security is the human element: employees, staff, customers. People can often lead to security vulnerabilities without ever intending to compromise a system. Hackers are skilled in using tools and techniques of human interaction to get people to disclose information, which is why it is important to keep the people who will have interactions with your network trained and educated to be security-minded.

Security awareness is normally spread through many different ways including: formal in person training, computer based training, emails, memos, bulletins, posters, etc. The goal of security awareness is to create an overall culture of awareness and protection.

These trainings and outreaches should be focused on teaching employees on the elements of security key to compliance and should be reinforced by a company's policies or security practices. An organization cannot reasonably expect their staff to follow policies if they have not been trained.

Security awareness and training is a good practice in principle, but often is ineffective. After a security training employees are often more alert for a few weeks, but soon fall back into the habits that made them vulnerable. Continual training is required and it should never simply be assumed that training completed years ago is still of any utility.

5. Case Studies

Brigham Young University's (BYU's) student Red and Blue Teams are a service organization that consists of 8 -10 students from the Cyber Security Research Lab (CSRL) at BYU. They regularly perform professional-grade penetration tests of departments within BYU and external organizations or corporations. Each assessment includes a negotiated security assessment proposal document consisting of the scope and rules of engagement, a permission memo and full verbose details of both procedural and technical measures. After the penetration test is complete the Red and Blue team provide a security assessment report consisting of a review of what transpired during the penetration test, what vulnerabilities were found, and the teams recommended mitigation techniques to better secure customer systems.

The Red Team is unable to disclose specific un-redacted information from penetration tests, however the following are based on real events:

5.1. Case Study I - Keyloggers

A keylogger is a device or software that is capable of recording the keystrokes of any user on the machine [11]. These devices are commonly a USB device that is connected to the end of a keyboard cable and is then plugged into a computer. Due to its small size, keyloggers can easily go unspotted if a person is not specifically looking for it. Using a combination of a keylogger and social engineering the red team was able to steal a system administrator's credentials.

The system administrators in the department are very aware of the concern of security on their own workstations. They have a policy in place that when any of them leave their workstation that they are to lock the computer. And if they are the last one to leave the system administration office they are responsible for shutting the door behind them and making sure it locks. Knowing this information, the Red Team planned accordingly.

As the system administrators had taken effective steps to physically limit access their consoles it was deemed impractical to attach a keylogger directly on a system administrator's workstation. The plan involved preparing for a system administrator to come to them. Using one of several open access computer laboratories the Red Team installed a physical keylogger onto one of the machines. They then engineered a realistic scenario so to allow them to social engineer a system administrator to use their credentials.

The scenario involved a relatively simple approach: disconnecting a network cable from the workstation with the planted keylogger. A member of the team then asked one of the system administrators for some help with their machine. Just as the red team assumed, the system administrator checked for the network cable, reconnected it and used his own credentials to log into the workstation to verify the connectivity of the box. This meant that the Red Team now had one of the system administrator's credentials.

In their report, the Red Team recommended that all systems should be checked regularly for keyloggers, especially public machines. While a well-intentioned idea, in reality it is often impractical to implement. For example, the aforementioned department has 100+ workstations that they are responsible for. All of these computers are open for use by any student taking a

class in their department, which means all the labs are publically accessible. The time allowed to check these machines with all the other responsibilities that the system administrators need to accomplish is too impractical to actually accomplish. However as we shall discuss a SNAP account would have significantly mitigated this attack vector.

5.2. Case Study II – Mimikatz

A different Red Team engagement, a request came from the department head to include physical security. Thus the scope reflected this and included extensive physical and social-engineering attempts.

As is typical, this penetration test began with footprinting and fingerprinting the department by gathering open source intelligence, running port scans and scanning for different vulnerabilities. The system administrators of the department had done a great job at locking down their department from the outside. They had a strong and well maintained firewall in place. Their website lacked vulnerabilities useful for us to gather information or exploit. After exhausting most of our options within the scope of an external attack the Red Team looked at how to penetrate their physical security.

Two teams where going to go in under two different pretenses. Team one decided to go in as "technical support" dressing in uniforms, carrying networking equipment, and having created a realistic looking fake work order. Team two went in as student customers to see if their might be any avenue of attack from that direction as well. Both teams had the same goal to be able to take advantage of any open computers by running mimikatz from a flash drive stealing credentials.

Mimikatz is a tool released in 2012 by Benhamin Delpy (aka gentilkiwi) that allows an attacker to harvest passwords from system memory on an unlocked console. It is worth noting that this tool is also usable on a remotely compromised system and is available in frameworks such as Metasploit as well as open sourced on github.

The teams knew system administrators of the department took on the role as help desk personal whenever an employee in their department was having computer trouble. The first team was successful at running Mimikatz on three consoles while posing as technical support personnel and retrieved credentials of a department director. The second team was repeatedly left alone in a room and were able to secure the credentials for department employees. In one instance, a member of staff was having technical difficulties which resulted in a system administrator visiting to investigate the issue. During his visit he left the team member alone in the room with the console unlocked. A few seconds was all that was required to plant the drive and retrieve the administrative credentials.

In their report, the Red team recommended that the staff and employees be better trained at habitually locking their computer whenever leaving it. However, this can only be so successful. As stated before a hacker only needs someone to make a mistake once and often hackers have the time and patience to wait for it to happen. The SNAP approach would have mitigated all administrative credential theft in both situations.

6. SNAP

SNAP (Secondary Non-Admin Privileged) is a security practice that will mitigate the theft of system administrator credentials. A SNAP account should have a different username and a different password separate from any administrator accounts. Having a SNAP account that can be used whenever working on a public workstation, or even on any task that does explicitly require administrative access adds another layer of protection for privileged credentials.

The underlying concept of a SNAP account is to divide a system administrator into two (or if necessary more) roles. Roles with a significant security separation are provided with a different account. This creates separation between the system administrator account and user accounts.

An example of this would be for a system administrator to keep their primary privileged account for access from trusted, administrative consoles. A second account would be available for working on less-trusted workstations such as those available to non-administrative users or even the general public.

In the case of a system administrator requiring administrative access to an untrusted computer, the administrator should be encouraged to add explicit permissions to their SNAP account from their regular administrative workstation on a temporal basis. Upon completion of the task, this access should be immediately revoked and if believed necessary, the SNAP account password rotated.

Implementing SNAP as a policy will help to ensure that administrators avoid the unnecessary use of privileged credentials on systems. This will help mitigate the risk of their credentials being stolen and used by malicious attackers.

6.1. SNAP Applicability

Depending on the organizational policy and infrastructure, SNAP may have a large or small risk mitigation impact. In cases where system administrators never use publically accessible systems or support users and customers locally, the risk-mitigation benefits of SNAP accounts are lessened. Yet as shown hereafter (use case C3), there are still

remote attacks that could compromise administrative credentials.

Universities are a good example of where it would be beneficial to use SNAP accounts. Many universities have student system administrators to help maintain their systems. While students may need to act as system administrators, they often also need to use public labs themselves for their schoolwork. They end up having to take on two conflicting roles: system administrator and student. This is a perfect example where SNAP should be used. Students will use the SNAP account whenever they are using the lab for their schoolwork, so if anyone steals their credentials at that time the damage is limited.

Using a SNAP account is also ideal for companies and organizations where any administrators with privileged accounts are also involved with assisting users on user located systems. Since system administrators are actively working with users and diagnosing system issues they have a lot of interaction with publicly accessible workstations. With this example, a SNAP account can add a level of protection to routine support tasks.

6.2. Implementation

The implementation of SNAP can be varied depending on the needs of an organizations. We will present three methods that we believe to offer some level of effectiveness.

6.2.1. Individual SNAP Accounts (i-SNAP). In this approach, each administrator has two accounts. This means a second unprivileged account being created for each system administrator and requiring unique passwords. This provides persistent SNAP accounts for administrators who regularly work on non-administrative workstations. It also maintains complete accountability with each user having their own, fully auditable account.

In a University setting, this may be an optimal implementation. In the case of student system administrators who are also taking classes, it allows the use of an unprivileged account when not directly involved in administrative duties, such as completing schoolwork or troubleshooting problems on lab systems. This can prevent a privileged account being stolen by a maliciously intentioned peer who might target them due to their position.

Other advantages of this approach include providing the ability for supervisors to audit accounts and see how, where and when system administrators are using administrative privileges versus their SNAP account. This allows for effective verification of appropriate account use as these accounts are used only when necessary and may be correlated to specific administrative tasks. This creates a level of accountability for the system administrators.

Having accountability also makes it easier to track the source if an account gets stolen. In the event of an administrative account breach, finding the source of the breach is simpler due to fewer account uses. Incident responders can thus respond more quickly in identifying the nature of the attack and in turn learn more about how their malicious attackers might be targeting them. All of this can lead to improved security posture against future attacks.

6.2.2. Shared SNAP Accounts (s-SNAP). A second way to implement SNAP is to have one SNAP account for all system administrators to share. While we do not encourage this due to the difficulty in providing accountability, it may provide a more achievable solution for system administrators that are less involved with the use of unfamiliar systems on a regular basis. This may be an effective means of using the extra security that SNAP provides while also helping decrease the amount of resources needed to implement it. Again, careful risk assessment is required to determine the suitability given the potential loss of accounting trails.

6.2.3. Temporary SNAP Accounts (t-SNAP). This implementation draws on the benefits of both previous options but may require supporting scripting to avoid causing delays in response times. It is most effective when system administrators use effective task management ticketing systems to manage support incidents.

In t-SNAP, a limited-privilege and limited-use account is created and activated in response to a specific incident. At the remediation of the incident, the account is disabled.

6.3. SNAP Use Case Examples

There are some specific types of attacks that can be effectively combated by adopting SNAP accounts, including (but not limited to): keyloggers, shoulder surfing, mimikatz and the forensic analysis of public workstations.

6.3.1. Keyloggers. As stated earlier, a keylogger is a device or software that is capable of recording the keystrokes of any user on the machine. Since keyloggers record the keystrokes of a user, they are often used to steal user credentials. Keyloggers share two common operations: 1) hooking into user input flow to receive keystrokes and 2) transporting the data to a remote location or storing it locally on the device or software's memory [12]. Any system administrator using a publicly accessible machine is vulnerable to

this attack, but if a SNAP account is used on all vulnerable public workstations then the account stolen will not have privileged system access and is thus of limited value to would-be attackers.

6.3.2. Shoulder Surfing. This technique is often employed as a social engineering tactic to gather information. Shoulder surfing involves watching or recording an individual as they type in their access code, password and/or PIN. It is a very simple, yet effective method of credential theft [13]. This technique can be even more effective when combined with social engineering to steal a system administrator's password. Most system administrators are very cautious of this type of attack, especially when at their own workstation. Companies will sometimes establish that clients and customers cannot walk past a line to the section where the system administrators are working for this specific purpose, but when a system administrator is helping a customer on a publically accessible workstation they become more vulnerable to this type of attack. Using a SNAP account, a system administrator can test these computers with less risk of their privileged credentials being stolen.

6.3.3. Mimikatz. Attacks such as pass-the-hash involve using privileged access to retrieve credentials from system memory (passwords may be in the form of reusable hashes, or clear-text passwords) [14]. Tools such as Mimikatz turn this attack into a simple mechanism for even unskilled attackers. As many users may possess local administrative credentials for their own workstation, administrators using public systems are at risk from both local malicious processes, and remote attacks during and shortly after their access. While such an attack could compromise any user using a Windows based host, it would be limited to the SNAP account rather than a privileged administrative account.

6.3.4. Forensic Analysis. A malicious attacker can use forensic analysis of a computer belonging to a system administrator to try and steal information. Forensic analysis is obtaining computer equipment such as hard-drives, DVDs/CDs, or sniffing network traffic flows while attempting to extract information that may be of use in attacking an individual or organization. While this is a relatively sophisticated form of attack, it is feasible for more determined attackers. There are good secure practices that, when implemented properly, can help protect against this type of attack. Password hashing changes the password to an undecipherable sequence of numbers and letters to avoid the password being stored in plain text. This in turn helps protect against the password being stolen through the forensic analysis of network traffic or media. Salting impedes the use of a form of attack that precalculates hashes by adding random data to each hash. Linux systems typically extend the password by 12 bits, enabling the same password to be stored in 4096 different ways and defeating rainbow table lookups [15]. Separating password hashes from other user information can protect passwords from live forensics (although offline media forensics will still show files). UNIX based platforms use shadow files with restricted permissions that help prevent intruders from reading hashes [15]. However, no system is perfect and there are always vulnerabilities. Misconfiguration and bugs in code can cause information that should be encrypted to be sent in plain text. A SNAP account should be used on these vulnerable public machines to help mitigate the risk by limiting an attacker's access.

6.3.5. Malware. System administrators are not immune from malware attacks whether they be generic, or targeted. Using a SNAP account for any workload that does not require administrative access – even for tasks such as accessing e-mail, may be an effective way of mitigating the damage of malware infections. Further research is required to establish the effectiveness of this as a malware mitigation strategy.

7. Conclusion

For the SNAP method to be most effective, SNAP accounts must be used whenever privileged access is not absolutely required – even on system administrator used workstations. This approach also requires that usernames and passwords used in SNAP accounts are not shared with or derived from those used in privileged accounts. These requirements necessitate effective training and administrative 'buy-in' to effectively reduce risk.

While system administrator roles may vary, all have some level of responsibility for an organization's infrastructure and information. Holding the 'keys to the kingdom' makes them a prime target for attackers. As such, they are effectively at the forefront in the battle of cybersecurity.

Administrative access provides attackers the opportunity to run significantly more attacks than is possible with unprivileged accounts. At the start of this paper, it was discussed how in 2013, 96% of critical vulnerabilities in Windows operating systems needed administrative rights. When it is considered that in this same study 100% of Internet Explorer's vulnerabilities required administrative rights; 91% of the vulnerabilities affecting Microsoft Office required these rights, and 100% of the critical remote code

execution vulnerabilities (which was found to have the highest impact) could be mitigated by the removal of administrator rights – the message is clear: Removal and protection of administrative rights is an essential cybersecurity strategy against today's threats [2].

We have found that the SNAP approach is an effective and efficient way to increase protection of administrative accounts. By protecting against inadvertent privileged credential loss it mitigates the risk of attackers gaining system-wide access to an organizations infrastructure. Due to its simple and low-cost implementation SNAP can fix multiple security exposures that organizations must face on a daily basis. We have demonstrated how SNAP can effectively defend against several attack vectors including keyloggers, shoulder surfing, Mimikatz and forensic analysis. We have also shown how this approach can enhance incident response in both attempted, and actual breach investigations. We believe that with appropriate training and enforced adoption, system administrators can use this method to better secure their privileged access and in turn the infrastructure for which they are responsible.

8. Acknowledgements

The authors wish to thank BYU's Office of Information Technology for their support of the Red Team and development of ideas in support of this paper.

9. References

[1] A. Wesley, the practice of system and network administration 2nd edition, 2nd ed. Boston, MA: Addison-Wesley, 2007.

[2] Avecto, "2013 Microsoft Vulnerabilities Study : Mitigating Risk by Removing User Privileges," 2013.

[3] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," Proceedings of 11th annual computer security application conference. pp. 241–248, 1995.

[4] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control," Proc. fifth ACM Work. Role-based access Control - RBAC '00, pp. 47–63, 2000.

[5] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," ACM Trans. Inf. Syst. Secur., vol. 4, no. 3, pp. 191–233, 2001.

[6] "What Network and Computer Systems Administrators Do," Bureau of Labor Statistics, 2014.

[7] "What is an administrator account?," Microsoft, 2015.

[8] "Pre-built administrator roles," Google, 2015.

[9] H. Setty, "System Administrator - Security Best Practices," SANS Inst. InfoSec Read. Room, no. May, 2001.

[10] D. R. Lindemann, "The Evolution of Authentication," FIDO Alliance, 2013.

[11] S. Moses, J. Mercado, D. Rowe, and A. Larson, "Touch Interfaces and Keylogging Malware," in Proceedings of the 11th International Conference on Innovations in Information Technology (IIT'15), 2015.

[12] C. a Wood and R. K. Raj, "Keyloggers in Cybersecurity Education," 2015.

[13] M. Allen, "Social Engineering - A Means to Violate a Computer System," SANS Inst. InfoSec Read. Room, 2007.

[14] S. Duckwall and C. Campbell, "Hello my name is Microsoft and I have a credential problem," in BlackHat 2013, 2013.

[15] M. Little, "Password Security: A Case History by Rober Morris and Ken Thompson (Bell Labs)," 2004.

