Global Information Assurance Certification Paper

Implementing the Critical Security Control: Controlled Use of Administrative Privileges

GIAC (GCIA) Gold Certification

Author: Paul Ackerman PAckerman@VacciNetLLC.com
Advisor:

Accepted: June 2016

Abstract

There is a plethora of information available to help organizations protect their cyber assets. So much so that it has become difficult for some to know where to begin. The Critical Security Controls from the Center for Internet Security address this issue by prioritizing defenses based on their ability to prevent attacks that are currently being used to compromise networks. In this paper, the author walks through actually implementing Critical Security Control 5: Controlled Use of Administrative Privileges. Capital and personnel resources, the level of effort, project steps and political challenges will all be addressed to help the reader successfully implement this control in their environment.

1. Business Value

There is a plethora of information available to information assurance professionals to help them identify what security controls could be in place to defend their environment. Between ISO 27002 (ISO/IEC 27002:2013, 2013), NIST 800-53 (NIST, 2015), the DHS Continuous Diagnostics and Mitigation Program (Continuous Diagnostics and Mitigation, 2015), the NSA Manageable Network Plan (Manageable Network Plan, 2015), the Australian Top 4, formerly known as the Top 35, (Strategies to Mitigate Targeted Cyber Intrusions, 2016), GCHQ's 10 Steps (10 Steps: Summary, 2015), the UK Cyber Essentials (Cyber Essentials Scheme, 2014), UK ICO Protecting Data guidance (Data protection self-assessment toolkit, 2016), PCI DSS (Requirements and Security Assessment Procedures, 2016), HIPAA (Health Insurance Portability and Accountability Act of 1996, 1996), the FFIEC Examiners Handbook (Information Security IT Examination HandBook, 2006), COBIT 5 (COBIT 5 for Information Security, 2012), NERC CIP (CIP Standards, 2016), Cloud Security Alliance (Cloud Controls Matrix v3.0.1, 2016), and ITIL (What is ITIL, 2016) there are literally hundreds of controls and, while all of them contribute to defending networks implementing every single control in any framework, all the time is unrealistic for most organizations. Likewise, there are numerous training companies that teach information security professionals how to attack and defend networks. Every blue team security professional at one point or another has had a conversation with their boss about going to offensive security training. The employee wants to go because hacking into stuff is fun, but the manager approves the training because he knows that in order to defend the organization, the staff needs to know how attacks work and how to recognize them when they occur. After all, it's difficult to stop what you can't see. But, it's not enough to know how different attacks work and what they look like because time and resource constraints make it extremely difficult to defend against all of them all the time.

We must prioritize defenses around which attacks are being actively used to compromise organizations. Just as defenders learn from each other, attackers' methods and tools tend to trend (find a source that confirms). Malware has evolved over time from

password stealing worms to botnets and banking Trojans and now ransomware. As new attack venues arise, they are typically very successful until defenses catch up and the cycle repeats. At any given time the quantity of resources that should be spent defending against a particular attack should be based on that attack's *current* likelihood to result in a breach of the organization. This is a moving target. The CIS critical controls try to address this issue by prioritizing defensive controls based on their ability to mitigate attacks that are being used to compromise networks today. By keeping the controls updated based on real attack and breach information and by evaluating defensive controls against their ability to stop those attacks, security leaders now have excellent guidance on where to focus their resources. While the critical controls provide the "what" to do, this paper takes a deeper dive into a single control and provides the "how" for Critical Security Control five: Controlled Use of Administrative Privilege (reference).

2. Water vs. Baking Soda

Critical security control 5 is defined as "The processes and tools used to track/control/prevent/correct the use, assignment, and configuration of administrative privileges on computers, networks, and applications. To understand why this particular control is so important, we must consider the risk it is designed to mitigate. A study by Avecto found that a whopping 92% of critical Microsoft vulnerabilities disclosed in 2013 can be mitigated by removing admin rights (Removing Admin Rights Mitigates 95% of Critical Microsoft Vulnerabilities, 2014). This data may be a few years old but older vulnerabilities are very commonly exploited. In fact, the average age of vulnerabilities used to compromise systems in 2015 was a little over six years so this data is still quite valid (Figure 1). As an administrator, an attacker can do a significant amount of harm on a compromised machine including things like:

- Replacing critical system files and registry settings to establish persistence and invisibility (think rootkit).
- Extract password hashes from disk and clear text password from memory (pwdump, Mimikatz).

- Installing root CA certificates to allow the system to execute additional malicious code or perform man-in-the-middle attacks.
- Install drivers that can interact with hardware (think keylogger).
- Change the passwords of all users on the system to lock out the owner.

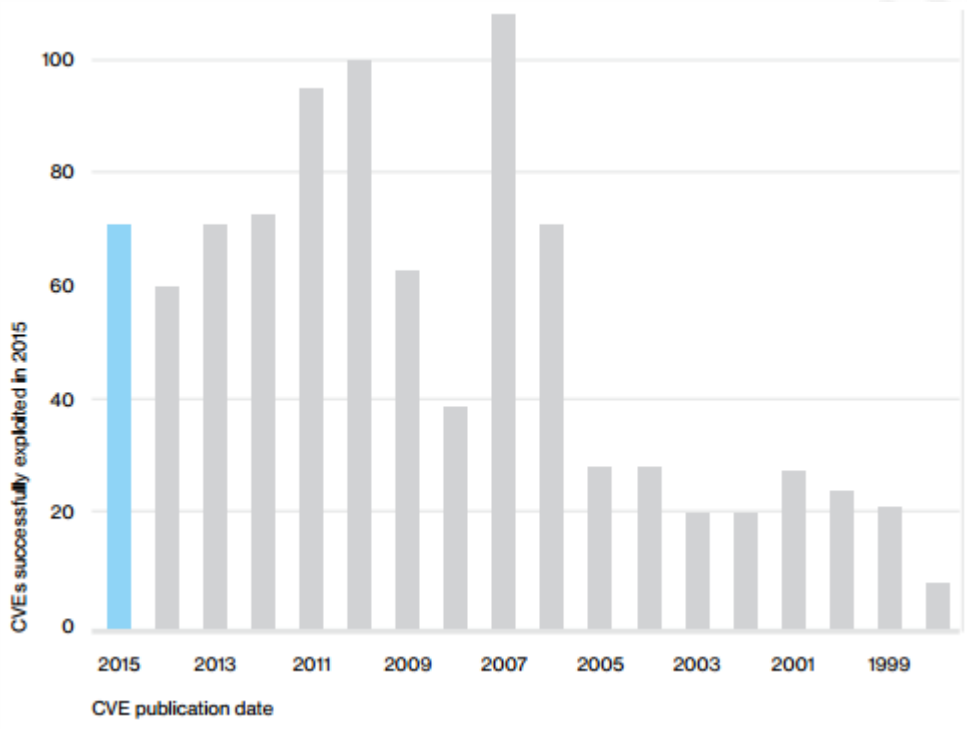


Figure 1: Count of CVEs exploited in 2015 by CVE Publication Date (http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/)

3. Project Planning

Reducing the usage of admin rights throughout the enterprise can be a fairly large project. Creating a project plan will help identify potential issues and ensure a success. The Standish group completes an extensive study of IT project successes and failures each year since 1994 called the CHAOS report. The 2015 report found that only 29% of projects were completed on time, within budget and that met the project deliverable

(Hastie & Wojewoda, 2015). In the original 1994 report, the group found that the top five factors for successful projects were:

- User involvement
- Executive Management Support
- Clear Statement of Objectives
- Proper Planning
- Realistic Expectations

(Frese & Sauter, 2003)

Similarly, the 2015 study, which included 50,000 projects around the world, found the following success factors:

- Executive Sponsorship
- Emotional Maturity
- User Involvement
- Optimization
- Skilled Staff

Interestingly, executive support and user involvement were both on the list in 1994 and 2015 implying these characteristics of successful projects aren't likely to go away anytime soon. For this reason, it is imperative that the project team fully understand the impact of the changes that will come from removing admin rights. This is where user involvement comes into play. The work breakdown structure created during the planning phase of this type of project should include scoping the project by discovering where admin rights are being used, determining least privilege requirements to identify where rights can be reduced, and identifying potential solutions and tools for situations where privileges cannot be reduced.

Begin the planning phase of the project by identifying where admin rights are currently assigned and/or in use in the environment. Identifying where rights are assigned can be accomplished by reviewing group memberships and wheel or sudoers configuration files, but log files should also be reviewed to catch missed or unexpected uses of those privileges (2.3 Configuring sudo Access, 2016). This can be accomplished using scripts that enumerate local administrators groups on workstations and servers and

by reviewing security event logs and `/var/log/auth.log` and `/var/log/btmp` on Linux machines to determine where the admin accounts are being used (Natarajan, 2011). Service accounts should also be reviewed to ensure they are running with the least amount of privileges necessary. Specifically, centrally managed software agents running on workstations are risky because administrators sometimes configure them to run under a local admin account rather than system or network service. In some cases, services are running on workstations using domain administrator privileges. Tools like Mimikatz can extract plain-text passwords for any logged-in account on the system – including service accounts (Mimikatz, 2016).

In an ideal world, every new system or application should be analyzed to determine what permissions are actually required for the application to function properly following the principle of least privilege. However, in the vast majority of cases of applications running on Windows, administrator rights are "required" by the vendor simply to avoid having to identify and/or instruct users how to configure the required NTFS, registry and other permissions and I.T. implementers often don't have the time, skill or motivation to do this work. This phase of project planning will likely be the most time-consuming part of the project since the information security team will need to work with the application subject matter experts to determine and document these requirements for each application. Since many of these requirements are not documented by the vendor, it will require some trial and error.

The planning phase should also include research into tools and solutions that can accommodate or mitigate circumstances where admin rights cannot be removed or reduced. While every organization is different, there are a few common strategies for mitigating the risk of admin rights that must remain in place. One common and easy to implement method is to create a "runas" account for those users that require admin rights and grant the rights to the runas account rather than the user's normal domain account. When the user needs admin rights, they can right-click and choose "Run as Administrator", enter their runas account credentials and complete the task. Workstations are most commonly compromised by websites or malicious email attachments. Using this

method, those activities are being run without admin rights, which greatly reduces the degree to which the workstation can be immediately compromised.

If the user is performing tasks that require admin rights very frequently or that don't work well under the runas context, another option is to create virtual machines that are used to browse the Internet and read email. This is an attractive option because the virtual machine can be very lean when it comes to applications and since the virtual machine isn't used for corporate applications, the OS, and installed applications can be patched without fear of the impact to internal applications resulting in a more secure browsing environment and reduced impact in case the VM is, in fact, compromised.

In some cases, users might be administrators because of a single legacy application that must run with administrative rights. In this case, a tool like BeyondTrust can elevate specific applications to run with admin rights without creating a second user account or second machine. The same benefits apply, however since the only application(s) running with admin rights are those specified in the BeyondTrust configuration. Once solutions have been identified for all business requirements, project execution can begin.

4. Tips and Risks

This type of project can be very politically challenging. Some users believe that it will be inefficient and frustrating for them to do their job if their admin rights are taken away (Phneah, 2011). While not necessarily true in all cases, the changes will most likely require users to do things differently whether that's using a separate account or a dedicated VM for browsing or some other modification to the way work is performed. There are tools that can be the exception to the norm, however. One example is BeyondTrust's powerbroker which works seamlessly to add an administrator token to specific processes with no impact to the way the user interacts with the system (Beyondtrust, 2016). With careful and thorough testing, this project can be executed with minimal impact to operations but there will most likely be a few unforeseen challenges that must be overcome in addition to the procedural changes mentioned above.

As mentioned in the CHAOS report, developing support from leadership will be crucial in overcoming these challenges without the risk of halting the project completely (Hastie & Wojewoda, 2015). To garner support from both leadership and users, begin by demonstrating the risk – live and in person. One option is to setup a laptop that can be taken to staff meetings and present a real-life compromise. Any type of compromise can affect the entire organization but ransomware is particularly effective to demonstrate since the inability to access files affects each user directly and is easy for many users to relate to. Once the risk has been communicated effectively, the next step is to communicate how you intend to proceed with the project with minimal operational impact. You must find and communicate the business value of getting the job done and take the project on with the attitude of supporting the business while improving security. Understanding the users' requirements is paramount to project success (Hastie & Wojewoda, 2015). In doing so, there may be ways to find small wins for the business. For example, BeyondTrust (Beyondtrust, 2016) may be deployed to allow a specific application to run properly but it can also be leveraged to enable users that were not previous administrators on their machines to do things like installing pre-approved applications from a pre-configured network share. In the author's experience, finding ways to provide business value will help offset the changes in procedure and perception that something is being "taken away".

During the discovery phase, project implementers may be surprised at how many users have administrative rights throughout the environment. Another risk to avoid is taking immediate, knee-jerk actions without performing a thorough analysis. This may result in a negative impact to operations or may create the perception that the project team doesn't care about the users' jobs. Either of these results can have devastating consequences for the project. To succeed, extreme care must be taken to ensure the minimal impact to normal business operations and to maintain the perception that business operations are equal to, if not more important than the changes that are being implemented.

5. Implementing the Control

5.1 The first component of control five is fairly involved. Control 5.1 includes minimizing the use of administrative accounts as discussed above and also auditing the use of privilege elevation. Begin by enabling success auditing of `Audit Process Tracking` and `Audit Privilege Use` via group policy. These settings are found in the Local Policies\Audit Policy node of the computer policy. Once auditing has been enabled, monitor for events 4648 and 4624 (How to Configure Auditing for Privilege Escalation, 2016). If you have a security event and incident management (SIEM) solution it should be fairly easy to monitor these events. One option to detect unauthorized use is to make a list of user accounts that have been authorized for admin rights and monitor for any events triggered using an account that does not exist on the list. That would imply an unauthorized privilege escalation has occurred. If the organization does not have a SIEM, Powershell scripts may be used as well, though this requires a higher level of effort. A scheduled task can be configured to trigger based on a windows event log. The task could then launch a Powershell script that retrieves the event from the security log and determines whether the conditions should trigger an alert. The alert could be sent via email. To read the most recent event 4624 from the security log, the command is simply:

```
$Event = Get-EventLog -LogName Security -InstanceId
4624 -Newest 1
```

Next, $Event can be checked against a list of authorized users:

```
$filename = "C:\scripts\authusers.txt"

$authUsers = get-content $filename

$authorized = $false

# Loop through each user and check to see if the event
contained an authorized user

foreach ($authUser in $authUsers){

  if ($Event.Message -like "*$authUser*" ) {

      $authorized = $true}
```

```
If $authorized = $false {

      Send-MailMessage -To "Gibson@myorg.org" -Subject
      "Unauth Priv Use Alert on $env.computername" -
      Body "$Event" -SmtpServer "smtp.myorg.org"-From
      NoSoupForYou@myorg.org -Priority High}
```

Control 5.2 specifies that automated tools should be used to inventory admin accounts and validate that each account is authorized. Tools like Viewfinity by CyberArk and PowerBroker by BeyondTrust can manage local administrator accounts and elevate privileges when needed. Enterprise password vaults such as Secret Server by Thycotic, Privileged Password Manager by Dell SecureWorks and Lieberman Enterprise Random Password Manager can also be used to centrally manage the passwords for local administrator accounts. As described above, Windows security event log can be monitored to detect real-time changes to the administrators group on workstations, servers, and privileged domain security groups. In addition to real-time monitoring, another approach is to use a workstation management suite like SCCM or Dell Kace's scripting engine to periodically report the administrators that exist on machines throughout the environment for review.

To enforcing control 5.3 - change default passwords on devices as they are deployed, create a policy that states the same and then perform periodic audits to ensure the policy is being followed. For common/standard devices, a template or checklist should be created that includes changing the default accounts. For new types of devices, a process should exist that enables an information security review of the device to determine the settings necessary for a secure configuration. Disabling default accounts and changing default passwords could be performed at that time. DHCP logging and switch CAM table monitoring can be useful in detecting unauthorized devices that have been attached to the network where 802.1x network-level authentication hasn't been deployed. Furthermore, wherever possible consider using RADIUS authentication to network-attached devices and enable local authentication only when the RADIUS servers

are unreachable. This is an additional safeguard that can be used to prevent unauthorized access if a default account is missed. Finally, a vulnerability scanner like Tenable Nessus or Rapid7's Nexpose can scan for default accounts on many types of common systems.

Control 5.4 deals with alerting when administrative groups are modified. Configuring alerts for accounts added to the domain administrators group or local administrators group on workstations can be done by monitoring windows security event logs on both local workstations and domain controllers. Event ID 4728 (formerly 632) is logged when a user is added to a global group. Event 4732 and 636 apply to local security groups. These events can be monitored with a SIEM or by using a scheduled task and Powershell script like the one used in section 5.1. A sample script to do this is included in Appendix A.

Implementing control 5.5 is also a fairly easy win. It specifies that there would be automated alerts on failed logins for administrative accounts. This is done by maintaining a list of administrative accounts and then monitoring event ID 4625 sub status code 0xC000006A using either a SIEM or Powershell script as described in other controls. Event 4625 is a failed login and the sub status code 0xC000006A is specifically a bad password. Login failures can also occur due to the account being disabled, locked or expired and will have different sub status codes (Windows Security Log Event ID 4625, 2016).

There are several third party solutions available to implement control 5.6 – Multi-factor authentication for administrative access. Vendors in this space include RSA, Dell, Duo, AuthLite and many others. And where multifactor authentication isn't supported, control 5.7 states that administrative accounts should require passwords of at least 15 characters. Active Directory group policy enables administrators to set password requirements but they apply to the entire domain. However, if the domain functional level is 2008 or higher, fine-grained password policies may be implemented instead and will enable administrators to configure more restrictive controls for admin accounts and less restrictive settings for other users. To enable fine-grained password policies, perform the following steps:

1 Open Active Directory Users and Computers and create a new global security group to which the password policy will be applied.

2 Add all regular (non-administrative) users to the new group.

3 Open the Active Directory Administrative Center, switch to tree view then go to System -> Password Settings.

4 Right-click on the Password Settings Container object and choose "New" then "Password Settings".

5 Configure the desired settings for regular users.

6 Click the "Add" button in the "Directly Applies To" section and select the global security group that contains all regular users.

7 Repeat steps 1 through 6 for administrative users.

(Roman P. , 2013)

Control 5.8 suggests that users should log in using unprivileged accounts and then elevate as needed using sudo or runas. This is a bit trickier than it seems in Windows. A list of users that are allowed to log in interactively can be configured in group policy but, unfortunately, this applies to both interactive logins (logging in and remote desktop) as well as runas logins. To clarify, one can limit the ability to log in using remote desktop via group policy but it is difficult to differentiate between a local login and a run-as login (femila, 2016). This configuration can be enforced however by changing the shell of the administrative accounts from explorer.exe to logoff.exe. The effect is that administrative users that attempt to login directly or via remote desktop will be immediately logged off (McLendon, 2006). One important safety tip is that by default only domain administrators are authorized to log in to domain controllers meaning you must either exclude domain administrators from this "default shell" policy or create other accounts to use to login to the domain controllers that are excluded from the "default shell" policy. Failure to do so will result in no one being able to login to the domain controllers. Another important tip is to avoid configuring this setting using local security policy. When group policy is used, the setting applies only to the user accounts that exist within the OU where the policy is

linked. However, when configured via local security policy, the setting will apply to all users. The default shell setting is located under User Configuration -> Policies -> Administrative Templates -> System -> Custom User Interface (Can I allow an admin account to use "Run As" but not logon to the desktop?, 2016).

On the Linux side of the house, sudo may be used to enable administrators to log in with individual named accounts and elevate to root for specific functions. The /etc/sudoers file is used to configure which commands a user is able to execute as root (Limiting access with sudo, part 1, 2001). Once the configuration is complete, the root account can be prevented from logging in via ssh by editing /etc/ssh/sshd_config and change the line:

#PermitRootLogin yes to

PermitRootLogin no.

(How to disable root SSH login and create sudo user, 2015)

If desired, the root account can also be prevented from logging into the console and via one of several methods (Disallowing root access, 2016):

1) Change the root shell to `/sbin/nologin` in the `/etc/passwd` file
2) Clear out the /etc/securetty file using the command "`echo > /etc/securetty`" then enable securetty support in the KDM, GDM and XDM login managers by adding the line:

```
auth [user_unknown=ignore success=ok ignore=ignore
default=bad] pam_securetty.so
```

to the following files:

```
/etc/pam.d/gdm
/etc/pam.d/gdm-autologin
/etc/pa    m.d/gdm-fingerprint
/etc/pam.d/gdm-password
/etc/pam.d/gdm-smartcard
/etc/pam.d/kdm
/etc/pam.d/kdm-np
/etc/pam.d/xdm
```

3) Disable logging in as root over ssh by editing the `/etc/ssh/sshd_config` file. Change the line:

```
#PermitRootLogin yes
```
to
```
PermitRootLogin no.
```

There are several options for implementing control 5.9 (Administrators shall use a dedicated, isolated machine for all administrative tasks). VMware VDI can be used either as the administrative machine or as the Internet machine. Alternatively, a browser and outlook client could be published via Citrix or VMware Horizon creating isolation from the administrator's workstation. In addition to the obvious benefit of limiting the impact of a compromise of an email/Internet workstation, there is the substantial and perhaps less obvious benefit of maintaining excellent control over the browser machine including plug-ins, patches, and version. Since the browser machine will not be used for internal applications, there is no dependence upon any particular version of Internet Explorer, java, flash or any other software that legacy internal systems may require.

6. Conclusion

By creating a well-thought-out project plan, approaching the project with a "least impact" strategy and implementing tools, scripts and policy changes to limit the use of administrative privileges, organizations can drastically reduce their risk profile. There are some additional controls that should be considered to enforce the configurations implemented above. For example, when creating separate accounts for admin use, controls should be in place to prevent those accounts from surfing the web and reading email. After all, the idea is to separate the attack vector (email and the Internet) from the administrative account. This can be accomplished using an outbound web proxy and by forcing integrated windows authentication for accessing email. Second, a process should be created to re-validate the continued need for any administrative accounts that were previously created for individual users. Third, try to enforce a separate password policy

for administrative accounts in a way that ensures users cannot use the same password for their regular and privileged accounts. This can be accomplished using fine-grained password policies and by performing regular audits using a script to dump hashes and look for duplicates. Finally, policies and processes must be created that define how admin accounts are to be used and how requests for admin rights are handled. These will vary by organization but a sample process and policies have been provided in the appendix for reference.

7. References

10 Steps: Summary. (2015, January 16). Retrieved from gov.uk: https://www.gov.uk/government/publications/cyber-risk-management-a-board-level-responsibility/10-steps-summary

2.3 Configuring sudo Access. (2016, July 19). Retrieved from redhat.com: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/2/html/Getting_Started_Guide/ch02s03.html

(2015, January 22). Retrieved from NIST: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjpv9rtw_HNAhWEHx4KHRH1A_QQFggcMAA&url=http%3A%2F%2Fnvlpubs.nist.gov%2Fnistpubs%2FSpecialPublications%2FNIST.SP.800-53r4.pdf&usg=AFQjCNEbHUQ5n2igvpokUKNT9vbBpshDTw&si

(2016, July). Retrieved from Beyondtrust: https://www.beyondtrust.com/products/powerbroker-for-windows/

Can I allow an admin account to use "Run As" but not logon to the desktop? (2016, July 4). Retrieved from AuthLite: http://www.authlite.com/kb/allow-runas-but-block-interactive-logon/

CIP Standards. (2016, July 13). Retrieved from nerc.com: http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx

Cloud Controls Matrix v3.0.1. (2016, June 6). Retrieved from Cloud Security Alliance: https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/

COBIT 5 for Information Security. (2012). Retrieved from isaca.org: http://www.isaca.org/COBIT/Pages/Product-Family.aspx

Continuous Diagnostics and Mitigation. (2015, November 6). Retrieved from DHS: https://www.dhs.gov/cdm

Cyber Essentials Scheme. (2014, June). Retrieved from uk.gov: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/317481/Cyber_Essentials_Requirements.pdf

Data protection self-assessment toolkit. (2016, July 13). Retrieved from ico.org.uk: https://ico.org.uk/for-organisations/improve-your-practices/data-protection-self-assessment-toolkit/

Disallowing root access. (2016, July 7). Retrieved from Redhat Customer Portal: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Security_Guide/s2-wstation-privileges-noroot.html

femila, e. a. (2016, July 15). *Implementing Least-Privilege Administrative Models*. Retrieved from TechNet: https://technet.microsoft.com/en-us/windows-

server-docs/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models

Frese, R., & Sauter, V. (2003, December 16). *Project Success and Failure*. Retrieved from UM-St Louis: http://www.umsl.edu/~sauterv/analysis/6840_f03_papers/frese/

Hastie, S., & Wojewoda, S. (2015, October 4). *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. Retrieved from InfoQ.com: https://www.infoq.com/articles/standish-chaos-2015

Health Insurance Portability and Accountability Act of 1996. (1996, August 21). Retrieved from US Department of Health & Human Services: https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996

How to Configure Auditing for Privilege Escalation. (2016, 06 25). Retrieved from SourceDaddy: http://sourcedaddy.com/windows-7/how-to-configure-auditing-privilege-elevation.html

How to disable root SSH login and create sudo user. (2015, January 26). Retrieved from TecAdmin.Net: http://tecadmin.net/disable-root-ssh-login-and-create-sudo-user/#

Information Security IT Examination HandBook. (2006, July). Retrieved from ffiec.gov: http://ithandbook.ffiec.gov/ITBooklets/FFIEC_ITBooklet_InformationSecurity.pdf

ISO/IEC 27002:2013. (2013, October 1). Retrieved from ISO: http://www.iso.org/iso/catalogue_detail?csnumber=54533

Limiting access with sudo, part 1. (2001, February 9). Retrieved from TechRepublic: http://www.techrepublic.com/article/limiting-root-access-with-sudo-part-1/

Manageable Network Plan. (2015, December). Retrieved from iad.gov: https://www.iad.gov/iad/customcf/openAttachment.cfm?FilePath=/iad/library/ia-guidance/security-configuration/networks/assets/public/upload/manageable-network-plan-guide.pdf&WpKes=aF6woL7fQp3dJicD7yZGy8Bfj75LPr7UkfDySy

Mandiant. (2014). *M-Trends 2015: A View From the Front Lines.* Retrieved 08 16, 2015, from https://www2.fireeye.com/rs/fireye/images/rpt-m-trends-2015.pdf

McLendon, C. (2006, May). *Managing a Custom Shell Using Active Directory*. Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/aa479087.aspx

Mimikatz. (2016, July 19). Retrieved from Offensive Security: https://www.offensive-security.com/metasploit-unleashed/mimikatz/

Natarajan, R. (2011, August 1). *20 Linux Log Files that are Located under /var/log Directory*. Retrieved from The Geek Stuff: http://www.thegeekstuff.com/2011/08/linux-var-log-files

Paul Ackerman, PAckerman@VacciNetLLC.com

PaintYourDragon. (2014, 6 14). *LEDBeltKit*. Retrieved 8 16, 2015, from GitHub: https://github.com/adafruit/LPD8806/blob/master/examples/LEDbeltKit/LEDbeltKit.pde

Phneah, E. (2011, September 20). *Improving security by removing admin rights not practical*. Retrieved from ZDNet: http://www.zdnet.com/article/improving-security-by-removing-admin-rights-not-practical/

Removing Admin Rights Mitigates 95% of Critical Microsoft Vulnerabilities. (2014, 02 21). Retrieved from Information Week Dark Reading: http://www.darkreading.com/attacks-breaches/removing-admin-rights-mitigates-92--of-critical-microsoft-vulnerabilities/d/d-id/1141356

Requirements and Security Assessment Procedures. (2016, April). Retrieved from PCI Data Security Standard: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf?agreement=true&time=1468451425569

Roman, J. (2014, 11 25). *Speeding Up Breach Detection*. Retrieved from Bank Info Security: http://www.bankinfosecurity.com/speeding-up-breach-detection-a-7604/op-1

Roman, P. (2013, May 29). *Step-by-Step: Enabling and Using Fine-Grained Password Policies in AD*. Retrieved from Microsoft TechNet: https://blogs.technet.microsoft.com/canitpro/2013/05/29/step-by-step-enabling-and-using-fine-grained-password-policies-in-ad/

Schwartz, M. (2014, 03 14). *Target Ignored Data Breach Alarms*. Retrieved from DarkReading: http://www.darkreading.com/attacks-and-breaches/target-ignored-data-breach-alarms/d/d-id/1127712

Strategies to Mitigate Targeted Cyber Intrusions. (2016, July 13). Retrieved from asd.gov.au: http://www.asd.gov.au/infosec/mitigationstrategies.htm

Suby, M., & Dickson, F. (2015). *The 2015 (ISC)2 Global Information Security Workforce Study.* Retrieved 8 16, 2015, from https://www.isc2cares.org/uploadedFiles/wwwisc2caresorg/Content/GISWS/FrostSullivan-%28ISC%29%C2%B2-Global-Information-Security-Workforce-Study-2015.pdf

What is ITIL. (2016, July 13). Retrieved from Axelos.com: https://www.axelos.com/best-practice-solutions/itil/what-is-itil

Windows Security Log Event ID 4625. (2016, 07 04). Retrieved from Ultimate Windows Security: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4625

Appendix A – Group Membership Alert

```
# GroupMembershipAdded.ps1
# Paul Ackerman
# Apr 2013
#
# This script is tied to a scheduled task which is based on event 4728 firing in the
# security event log in order to send an email alert when critical AD group
memberships
# are modified. Event 4728 specifically occurs when a member is added to a group.

# Get the most recent event object from the security log
$Event = Get-EventLog -LogName Security -InstanceId 4728 -Newest 1

# Build an array of groups we want to monitor either directly or from a text file
#$groups = @("Domain Admins","Enterprise Admins","Schema Admins")
$filename = "C:\scripts\groups.txt"
$groups = get-content $filename

# Loop through each group and check to see if the event contained that group
foreach ($group in $groups){
 if ($Event.Message -like "*$group*" ) {

   # Event contains a critical group - fire an email alert
   $MailBody= $Event.Message + "`r`n`t" + $Event.TimeGenerated
   $MailSubject= "***ALERT*** A member was added to a critical Security Group"
   $SmtpClient = New-Object system.net.mail.smtpClient
   $SmtpClient.host = "your mailserver"
   $MailMessage = New-Object system.net.mail.mailmessage
   $MailMessage.from = "AD_Audit@yourdomain.org"
   $MailMessage.To.add("SecOPS@yourdomain.org")
   $MailMessage.IsBodyHtml = 0
   $MailMessage.Subject = $MailSubject
   $MailMessage.Body = $MailBody
   $SmtpClient.Send($MailMessage)
 }
}
```

Appendix B. Sample Admin Rights Acknowledgement Form – for non-IT users

This purpose of this document is to define the use of local administrator accounts on Windows workstations in order to mitigate the risks associated with these accounts

while providing necessary access for staff to complete job functions. On workstations, if users are logged in reading email and browsing the internet with local administrator privileges it is not only significantly easier for an attacker to compromise the workstation but the extent to which the workstation can be compromised is also much higher. By using a separate account with a different password for administrative tasks, users can significantly help protect the organization.

Responsibility:

1. Local administrator privileges do not imply that the user may perform any desired task on the workstation. Users must follow approved processes for software vetting and adhere to existing policies regarding subverting security controls and acceptable use.
2. Users with administrator rights are not permitted to grant administrator rights to other users or accounts. All permissions assignments must be requested via a service desk ticket.
3. If users choose to store the password for an administrative account electronically, it must be stored in the enterprise password vault. Passwords may also be stored on a company-provided secure flash-drive, as a backup.
4. The password on the administrative account must never be the same as the password for any other account either within or outside of the organization.
5. Users may elevate privileges when needed by utilizing the "run-as" function or by simply entering the username and password for the administrative account when prompted. Users should not log in to the workstation with an administrative account.

I understand the guidelines listed above regarding the use of an administrator account and agree to follow them at all times. I understand further that a failure to follow these guidelines may result in mandatory security awareness training or the loss of the administrator account which may reduce my ability to perform certain tasks without IT assistance.

Printed Name: _____

Signature: _____

Date: _____

You must report any performed or witnessed violations of this policy immediately to the InfoSec team with any supporting documentation that may be necessary and cooperate with any investigation that may commence as a result of the violation.

Appendix C. Sample Admin Rights Policy for IT users

Purpose:

Paul Ackerman, PAckerman@VacciNetLLC.com

The purpose of this policy is to define the use of local admin accounts on Windows servers and workstations in order to mitigate the risks associated with these accounts while providing necessary access for staff to complete job functions.

There are two main security-related challenges surrounding the use of local administrator accounts on workstations and servers. The first is the risk to the host itself. On workstations, if users are logged in reading email and surfing the internet with local administrator privileges it is not only significantly easier to compromise the workstation but the extent to which the workstation can be compromised is also higher. A machine infected under a local administrator account usually requires a rebuild while one infected as a non-admin user can be cleaned and placed back into service. Using a local administrator account for normal activity directly risks all other local accounts that exist on the workstation as well as any domain accounts that are logged in to the workstation such as service accounts used by antivirus, security incident and event management, inventory and others. These accounts are all accessible by a member of the local administrator's group and can be retrieved from memory in a few seconds if a user with admin rights clicks on a malicious link in an email or visits the wrong webpage.

The second main challenge is the re-use of accounts. The objective is to minimize the number of local and domain accounts that have privileges across many machines. In many organizations, local administrator accounts on workstations have the same password throughout the organization or at least within departments. This fact makes it extremely easy for the attacker to pivot throughout the organization and compromise other workstations that have access to protected systems that the originally infected machine may not have access to. These accounts are targeted by attackers and make the spread of an attack throughout the network extremely easy, fast, and difficult to detect.

Scope:

This policy applies to anyone that is authorized to configure administrator rights as well as anyone that receives administrator rights for any company-owned workstation or server.

Definitions:

1. Domain Account – This is a user's regular Active Directory domain account that is used to login to windows workstations. It has no administrative rights on workstations or servers and is used to browse the internet, check email and perform other non-admin tasks. It will be named using the format: FirstNameLastName

2. Domain Administrator Account – This is an Active Directory domain account that is a member of either the Administrators, Domain Administrators group or the Enterprise Administrators group in Active Directory. This account may be used to perform administrative tasks in the domain. It will be named using the format: FirstnameLastname-DA. The password length on this account must be at least the minimum length specified in the domain password policy + 7 characters.

3. Domain Run-as Account – This account is an Active Directory domain account that will have administrative rights on an end-user computing devices such as a workstations. It is designed to be used to elevate privileges when necessary utilizing

the right-click run-as functionality in Windows. It will be named using the format: FirstNameLastName -RA. The password length on this account must be at least the minimum length specified in the domain password policy + 4 characters.

4. Domain Server Admin Account – This account is an Active Directory domain account that will have administrator rights on servers. It may be used to RDP to a server and may be used interactively. This account should not be used to login to workstations at all. It will be named using the format: FirstNameLastName -SA. The password length on this account must be at least the minimum length specified in the domain password policy + 2 characters.

5. Local Administrator Account – This is a local user on a specific device that is a member of the local administrators group. This includes built-in administrator accounts and any additional local user accounts that have been created and added to the administrators group on the device. It will be named using the format: FirstNameLastName-LA.

6. Malware –short for malicious software is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. It can appear in the form of executable code, scripts, active content, and other software.

Responsibility:

6. Persons that configure any of the three administrative accounts {Domain Run-as Account, Domain Server Admin Account, Domain Administrator Account} must always obtain the proper authorization prior to granting privileges. Refer to the "Local Admin Privilege Request Process" for the Windows admin rights authorization process. Refer to the process: "Windows Local Admin Privilege Request Process.vsd"

7. If users choose to store passwords, they must be stored in the enterprise vault and users must use the account with the highest impact to login into the vault. Information Security will determine which account a user should use to access the vault as new privileged accounts are created. As an alternative, users may also use the company provided secure flash drive to store passwords.

8. Persons that configure administrator rights on workstations:
    a. Must obtain approval to create the account by the requester's manager and the Information Security Officer (ISO) or delegate.
    b. Local administrator accounts used for imaging and tech-support will have randomized passwords so they are unique on each workstation throughout the organization.
    c. Passwords for local administrator accounts on workstations used for imaging and tech-support will be stored in the enterprise password vault.
    d. Desktop Support will enter new workstations into the vault as they are built.

e. Domain run-as accounts will be created for users that have been approved for admin rights on a workstation.
    i. GPO restrictions will be used to prevent these accounts from interactively logging into servers. (2nd local group called RunAs that is a member of the local administrators group; use GPO restricted groups to control members of the runas group; restrict logins for that group; rdp, local, service, batch job rights, default shell)
f. New service accounts must be granted the minimum privileges necessary. Any request for a new service account with local administrator privileges must have a documented technical justification for those privileges.
g. Service accounts used on workstations must not be used on servers to prevent an attacker from immediately pivoting from a compromised workstation to a server.
   Existing service accounts at the time of this policy must be assessed to have their privileges reduced if possible and to ensure they are not being used on servers as well.

9. Persons that configure administrator rights on servers:
    a. Local administrator accounts used during the build will be renamed and will have randomized passwords so they are unique on each server throughout the organization.
    b. Passwords for local administrator accounts on servers will be stored in the enterprise password vault.
    c. Persons responsible for building new servers will enter those servers into the vault as they are built.
    d. Domain Server Admin Accounts will be created for users that have been approved for administrator rights on Windows servers.
    e. AD Security Groups will be created and added to the local administrators group on servers. The user's domain server admin account will then be added to the AD group to be granted privileges to the servers rather than being added to the local admin group directly.
    f. Domain server admin accounts will be added to the "No_Internet" AD group to prevent Internet access via the web proxy. Exceptions will be granted based on system-specific requirements.
    g. New service accounts must be granted the minimum privileges necessary. Any request for a new service account with local administrator privileges must have a documented technical reason for those privileges.

10. Users that receive Domain Administrator rights in Active Directory:
    a. These accounts must be used only for tasks related to administration of the domain itself included but not limited to group policy, users and computers, sites and services, etc.
    b. These accounts must not be used to login to end user computing devices

c. These accounts must not be used to login to servers other than domain controllers.

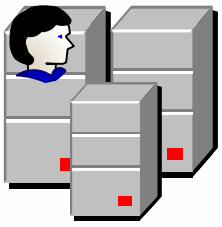
11. Persons that receive administrator rights on workstations:
    a. The password on the domain run-as account must never be the same as the password for any other account.
    b. Users may elevate privileges when needed by utilizing the "run-as" function. Users should not log in locally with a domain run-as account.
    c. Service accounts used on servers must not be used on workstations to prevent an attacker from immediately pivoting from a compromised workstation to a server.
    d. Local administrator privileges do not imply that the user may perform any desired task on the workstation. Users must follow approved processes for software vetting and adhere to existing policies regarding subverting security controls and acceptable use.
    e. Local administrator privileges do not imply that the user may grant other users administrator rights. All permissions assignments must be requested via a service desk ticket.

12. Persons that receive administrator rights on servers:
    a. The password on the domain server admin account must never be the same as the password for any other account.
    b. Local administrator privileges do not imply that the user may perform any desired task on the server. Users must follow approved processes for software vetting and adhere to existing policies regarding subverting security controls and acceptable use.
    c. Local administrator privileges do not imply that the user may grant other users administrator rights. All permissions assignments must be requested via a service desk ticket.

User Account Summary

A user may have up to 4 different AD accounts depending on what rights are needed in the organization. The following is a summary of the four accounts. The example user's name is Paul Ackerman

Domain Account: Ex: paulackerman 	This account is used to login to the user's primary workstation to perform everyday tasks such as reading email, browsing the Internet and accessing applications. It has no admin rights. Threat Level: Critical x Impact Level: Low = Risk Level: Medium

Domain Server Admin Account Ex: paulackerman-SA 	This account is used to RDP into servers. It is a local admin on specific servers. It should never be used on a workstation other than to RDP to a server. Internet access is prohibited on servers by policy with exceptions. Threat Level: Low x Impact Level: Medium = Risk Level: Medium
Domain Run-as Account Ex: paulackerman-RA 	This account is used to run applications as an admin on workstations. It should not be used to login to a workstation locally or through RDP but through the right-click run-as function in windows to elevate privileges as needed. Occasionally, there may be a task that does not function with run-as such as launching control panel to uninstall software. Local login is permitted for only these functions that cannot be performed using run-as. Threat Level: Medium x Impact Level: High = Risk Level: High
Domain Administrator Account Ex: paulackerman-DA 	This account is used to administer the domain/forest. It may be used to login to Domain Controllers interactively either locally or through RDP. It may also be used in a run-as function on select workstations to elevate applications that administer the domain such as the Group Policy Management Console, Active Directory Users & Computers, Powershell, Active Directory Sites & Services, etc. It should not be utilized to perform functions as a local admin on workstations or servers but should be used only for domain tasks. Threat Level: Medium x Impact Level: Critical = Risk Level: Critical

I understand the guidelines listed above regarding the use of an administrator account and agree to follow them at all times. I understand further that a failure to follow these guidelines may result in mandatory security awareness training or the loss of the administrator account which may reduce my ability to perform certain tasks without IT assistance.

Printed Name: _____

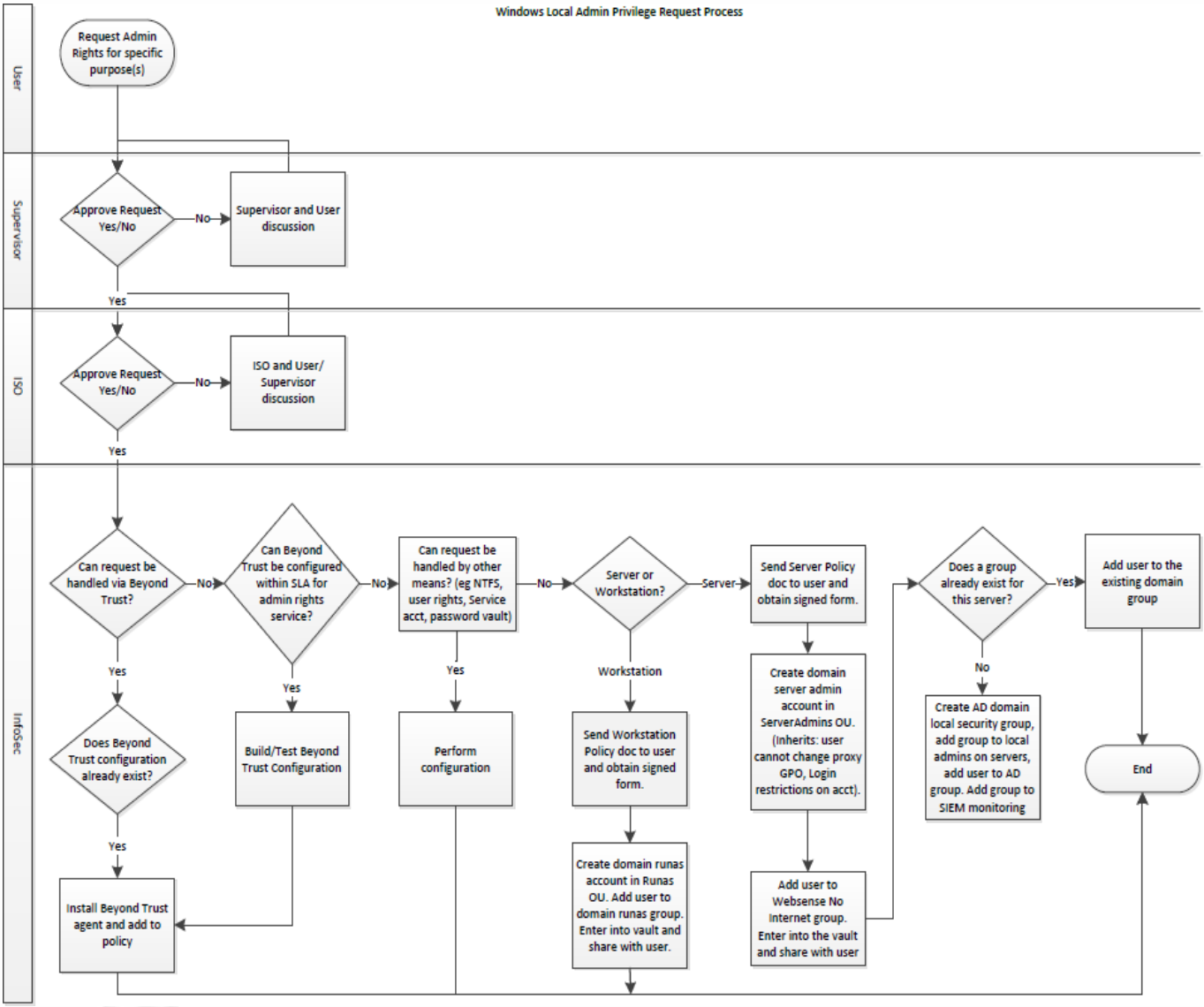
Signature: _____

Date: _____

You must report any performed or witnessed violations of this policy immediately to the Information Security team with any supporting documentation that may be necessary and cooperate with any investigation that may commence as a result of the violation.

Paul Ackerman, PAckerman@VacciNetLLC.com

Appendix D – Sample Admin Request Process



Paul Ackerman, PAckerman@VacciNetLLC.com