International Journal of Security and Its Applications
Vol. 12, No. 2 (2018), pp.45-58
http://dx.doi.org/10.14257/ijsia.2018.12.2.05

Towards a More Secured Solution in RDP: On-Demand Desktop Local Admin Rights

Manal Al-Sharrah[1] and Mohammad Alkandari[2]

[1]*Kuwait Oil Company*
[2]*Department of Computer Engineering, Kuwait University*
[1]*mhsharrah@kockw.com*, [2]*m.kandari@ku.edu.kw*

Abstract

One of the primary security threats against organizations is compromising administrative credentials and uncontrolled access to major systems by privileged users. The number of security attacks towards sensitive industries, such as oil and gas, is increasing rapidly and experiencing destructive and data breaches involving stolen user accounts information. Failure to prevent such security breaches would cause the enterprise major failures such as: damaging the enterprise image, exposing sensitive data, causing system failures, and even putting the national security on risk. One way to protect the user credential in an enterprise is to have a managed process for privileged user's access. This paper proposes and implements an on-demand solution to effectively manage, monitor, and control desktop local admin access for specified assets. The proposed solution has been tested and implemented at Kuwait Oil Company. The solution automates local admin service requests' workflow, develops on-demand admin Web interface, integrates with KOC Active Directory to authenticate and authorize users, keeps logs and audit reports, and implements a real-time interactive dashboard to monitor any suspicious activities and lower any security vulnerabilities.

Keywords*: Privileged access management; Admin account; Security; Identity management; Access control*

1. Introduction

Controlling privileged user access is becoming a vital step in any organization due to the increased risk of attacking. Currently, the number of enterprises suffering from security threats and attacks is dramatically growing, especially in sensitive industries such as: oil and gas, energy, defense departments, financial companies, *etc.* According to a survey which was conducted in 2015 [1], a tool was used to evaluate the organizational cyber security challenges faced in the energy sector, it was reported that 77% of the organizations had faced an increase in successful cyber-attacks in the past 12 months. Attackers are now using advanced methods to penetrate into vulnerable systems. Although the majority of the organizational attacks are often considered external, the fact that there is a possibility for internal threats is equally terrifying. Many organizations designed their systems to trigger external attacks, but very few actually monitor the internal activities in detail, especially for legitimate users. On August 15th of 2012, the computer network of Saudi Aramco Company was hit by Shamoon virus. This attack resulted in wiping the data in 35,000 computers in the company putting one of the major oil suppliers in the world at a risk. The company went offline for days to prevent the replication of virus over the network. The company reported that the main reason for this attack was due to a scam email in which one of the technician employees has opened it, hence allowing the hackers to get into the system [2]. After this incident, many

Received (October 30, 2017), Review Result (January 23, 2018), Accepted (February 1, 2018)

ISSN: 1738-9976 IJSIA

organizations created external security controls and awareness to avoid such threats. However, attackers are starting to target privileged users in order to get their administrative credentials to hack into the system. Therefore, it's very important to consider possible solutions to secure and monitor privileged account's credentials to avoid data breaching. Failure to prevent such data breaches will cause harmful system damages, destroy the image of an enterprise, scarify sensitive data to the public, cause interruptions in business operations, *etc.* Hence, without an effective solution to manage privileged access, the enterprise will not only face security attacks risks, but it will also suffer from difficulties to meet some standard compliances such as ISO/IEC 27000, which is a group of standards that helps organizations in securing information assets [3].

Privileged Access Management (PAM) is a solution in which privileged access is restricted in organizations in order to segregate the privileged accounts use to reduce the risk of stealing those credentials [4]. Gartner, Research Company providing technology insight for IT leaders defines PAM as "technologies help organizations protect critical assets and meet compliance requirements by securing, managing and monitoring privileged accounts and access. Gartner has renamed PAM from privileged account management, as it used to be, to privileged access management" [5]. One of the main challenges for organizations today is to get access to different sensitive recourses. Such privileges can be escalated within the environment if the access was unauthorized. Attackers can find different tools or even techniques, such as pass the hash and pass the ticket, to obtain domain account credentials. Once this unauthorized access is granted, it will be too hard to determine such internal attacks since the process of granting an access is not monitored. As per Gartner market research in [5], it's estimated that by 2017, there will be more regulations regarding controlling privileged access and this in turn, will lead to a rise of 40% in penalties forced on organizations that apply inefficient PAM controls.

The aim of this paper is to control and secure the organizational environment and give the organization the ability to monitor all privileged accounts in order to know who is doing what, at what time and identify the place. The proposed solution will give the ability for users to request on-demand local admin access right on desktop computer with monitored time access. The process will be fully automated, with no human interaction, by providing a Web self-service interface for the end user to request the admin access. Therefore, this will give an overall picture of how administrative accounts are being used. In addition, the proposed solution adds a layer to authenticate administrative users that have access to different domain joined computer and it makes sure that this access is only valid for a specified time. Applying this strategy in any organization will restrict the use of privilege accounts and it will also provide a detailed insight on how the privileged accounts are being used in a specified computer.

2. Motivation

The motivation behind this project was to find a way to manage access in privileged accounts by providing on-demand access for remote desktop connection. Organizations tend to give admin rights for users without keeping track of what will happen later after they finish their required tasks, leaving this process to be based on manual work with many security vulnerabilities. Thus, most applications support team members will end up having unmonitored and permanent local admin privileges on desktop computer since this access will be forgotten to be revoked. In addition, most of the work to manage an application or to maintain anything on desktop computers is mostly carried after the working hours for organization. Hence, a process is needed to be established to support on-demand requests. From those challenges, the need to implement a PAM solution was highly beneficial. Since Kuwait Oil Company is one of the leading and critical companies in Kuwait (KOC), this solution is tested and implemented based on KOC systems and

standards to enhance the security and control the process of managing desktop local admin access rights.

3. Contributions of this Project

1. Automate and integrate local admin service request workflow.
2. Develop on-demand admin Web interface.
3. Provide a secure way to authenticate and authorize the end users.
4. Control the process of requesting privileged local admin access accounts.
5. Implement a logging and audit reporting dashboard.

4. Organization of this Paper

The organization of the remainder of the paper is as follow: section 5 surveys the literature and lists similar conducted works, studies, and research related to privileged access management. The requirements of the proposed solution of implementing on-demand desktop local admin right are explained in section 6, while the overall process is described in detail in section 7. The design phase is described it section 8 along with the hardware and software required technologies that are needed to build the system. Section 9 shows a high-level architecture of the solution and it explains why this architecture is unique compared with the traditional software architectures. Section 10 explains the implementation phase along with the workflow that integrates different system components, and automates the whole process to be executed without any human interaction. In addition, the details and samples of the generated logs, report, and dashboard are shown in the same section as well. Finally, the paper concludes with section 11; a summary and potential ideas for future work to further enhance this system.

5. Literature Survey

There are many approaches in managing admin accounts and user access. However, in large organizations, the process of managing those accounts will become very complex and hard to handle. This is due to the fact that in large size organizations, the number of employees is big and spread among multiple places. Furthermore, there will be different heterogeneous applications that are based on different types of systems with different types of controls. One of the simple ways to manage administration accounts and user access right is to use the Role-Based Access Control (RBAC). This approach organizes permissions based on roles, not directly associated per user. Hence, multiple roles will be created, each with a specific admin privileges, and then users will be assigned to these roles [6]. This approach allows a good segregation of responsibilities. On the other hand, the implementation of this approach is limited to single systems. The paper in [7], overcomes this limitation by implementing an Enterprise Role-Based Access Control (ERBAC) approach that is based on the RBAC model. The ERBAC approach is deployed in the commercial security tool SAM Jupiter which enables automation in administration processes. This approach has proven that it can reduce the number of access roles dramatically if some parameters are added to the initial approach.

The paper in [8] shows how important it is to merge the best features of RBAC and combine them with some attribute parameters to provide an efficient access control for distributed systems and dynamic applications. This approach became a popular practice sine RBAC based systems will not support or provide any

flexibility for dynamic attributes such as considering the time of the day, which in turn can be taken as an important attribute to determine the end users' permissions. Therefore, Attribute-Based Access Control (ABAC) focuses on the fact that attributes and rules could either change RBAC approach or it could enhance it in terms of adding flexibilities for dynamic nature based systems. The additional attributes in ABAC based systems can specify the conditions to know a certain access is approved or denied. For example, an organization might grant one of its regular employees an access during the working hour. The same type of access might also be given for a supervisor or auditor who has a management privileges. Therefore, this approach is more flexible than the regular RBAC model because it does not separate the roles for related sets of subject attributes. In addition, those attributes and roles can be changed quickly in order to accommodate any changes as opposite to RBAC. The researchers in [8] argue that although ABAC brings a lot of advantages, there are still few disadvantages that need to be considered. RBAC trades up when it comes to developing the role structures efforts, hence allowing an ease process in the administration and user access review. On the other hand, ABAC trade off since analyzing or changing user permissions can be very difficult. The paper also lists three RBAC-A approaches based on the relationship between roles and attributes. The first approach is "dynamic roles" which was demonstrated in the example stated above. The second approach is attribute-centric in which a role is not based on a group of permissions, like RBAC, but the name of an attribute itself is called role. One of the drawbacks for this model is the requirements of many attributes to be added. In contrast to this approach, the third approach, role-centric, adds attributes to constrain RBAC it, which it helps to reduce the given permissions to the users.

Another interesting work in [9] is also based on RBAC, which focuses on providing efficient access control on Web servers. Most of the current methods to access control on Web servers are directly linked as per user identity. Therefore, those methods introduced some limitations in scaling to large organization systems. To overcome this problem, the researchers in [9] combined Web technique along with RBAC to deploy an effective and secure solution for large organization. This is done according to two different architectures: user-pull and server-pull architectures. In the first approach, the end user pulls the associated roles with his/her account from the role server. Those roles are stored in the local user's machine and then they can be presented to different Web servers. On the other hand, each Web server in the server-pull architecture pulls the end user's roles from the role server on demand. Both architectures are secured with secure cookies and smart certificates since they are Web based, but both have different pros and cons. Since in the user-pull architecture the secure credentials are stored in the local end user's machine, the end user can use those roles in many different sessions and Web servers until they expire; hence, this architecture will increase reusability as opposite to the server-pull architecture in which the roles are assigned to the user in each session. However, the latter will have long freshness of the obtained roles. Therefore, each architecture can be exploited in different context or application as long as there is a systematic way to manage the end user access and protect its associated account details and rights.

Surveying the Literature has revealed to us that there are different methodologies and approaches in which an organization controls and manages privileged accounts. Most approaches are based on RBAC model or depend on it relatively. The paper in [10] defines a unified approach for RBAC by combining different ideas from RBAC models, commercial products, and research methodologies. It organizes the unified model into four stages of increasing functional capabilities. The first stage called flat RBAC in which it represents the vital aspects of RBAC. The unified approach

model requires that the assignment type between roles and users must be many-to-many. This means that a user can be assigned to many rules and at the same time, rules can also be assigned to many users. The second stage is the hierarchical RBAC where it adds a new requirement based on the previous stage. As the name implies, this stage requires that permissions shall be organized according to their hierarchy order. For example, a senior employee must obtain all the rules assigned to his juniors' staff. The next stage is constrained RBAC in which the added requirement helps in enforcing separation of duties. The techniques in this stage assets in decreasing the likelihood of fraud and any other system damages. The last stage is symmetric RBAC in which it requires permission role review.

One can notice that over the past years, there are significant progresses made and efforts to define, model, and implement techniques to manage the user access and privileged accounts. Nowadays, such implementations are very critical to secure the organizations. The book in [11] focuses on access control systems from security perspectives. Managing the process of user access and permissions all depends on the identification and authentication of the user in the first place. Hence, the aim is not only to monitor privileged accounts, but also to make sure that only authorized users can access a computer system or any service in the organizational network. In this paper, the proposed approach takes care of managing on demand user access while taking care of the system from the security's perspectives.

6. Requirements Phase

The functional requirements of this paper are to develop a managed and secure process to do the following:

1. Deliver an on-demand and automated solution to manage the process of requesting full privileged local admin account for desktop computer.
2. Control this process by limiting the time of the local admin access in specified asset, computer, instead of limiting the user's access or privileged.
3. Develop on-demand Web self-service on the intra-net of the organization in order to allow the end user to request a remote access to a specific computer at any time, even after the working hours.
4. Provide security check to authenticate and verify the user before accessing any remote asset by integrating the solution with Microsoft Active Directory (AD) for authentication and verification.
5. Provide second level of security by making sure that only the verified and eligible users who can utilize this solution are the ones who have access on its interface and are also members of the privileged admin users' group in which it's monitored and managed by the Information Security Team of the organization.
6. Give the local admin user the freedom to specify the duration of required time to access a specific desktop computer to finish a certain task.
7. Notify the local admin user by sending emails to his/her account about any alerts, changes, and messages such as: access granted, access denied, error messages, duration of time is about to end, *etc.*
8. Offer flexibility for the local admin user in the duration of access to the remote desktop computer by giving an option to extend the duration of access time if the required task is not finished within the pre-specified time duration when the request was created. A justification is needed to extend the request to further enhance the security for the solution.

9. Provide logs and auditing reports that keeps track of everything in the system: action time, work order, requester name, requester user name, computer name, justifications, duration of access, status of access, messages, time the request is created, time of the request modification, status of the request if it's extended in time or not, status of access it's revoked or still valid, *etc.*
10. Implement a live and interactive dashboard to provide live monitoring tool for the whole process. The dashboard will also show all the information of the report logs. The user will be able to select the period of time, one week for example, to know what happened exactly in that week. The dashboard will also provide the ability to view a performance monitor to the average processing time of user requests.

7. Overall Process

To understand the overall process of the proposed solution, Figure 1 shows the main components of the process. First, the admin user will use the Web self-service to access the on-demand portal. Through the Web-portal, the admin will create a request to get a remote access on a certain desktop computer by entering the PC name, access justification, and select the task duration to make sure that the privileged access will be monitored and only available for specific time. It's important to notice that this solution only provides remote desktop access to computer, not servers. In addition, the aim of this solution is to manage the most privileged accounts which have full access admin right by making sure that those accounts maintain this privileged in limited time only. Hence, the solution will limit the time of the privileged accounts, not the access type. The next step in this process is to authenticate and authorize the admin user through integration with AD. The user will be asked to put his/her credentials for authentication and to make sure that he/she is authorized to get full local admin privileges. If the conditions are not met, the user access will be denied; otherwise, the user will be granted a remote access to the specified computer and the session will start. Once the specified time duration ends, the session will end and the access will be revoked.

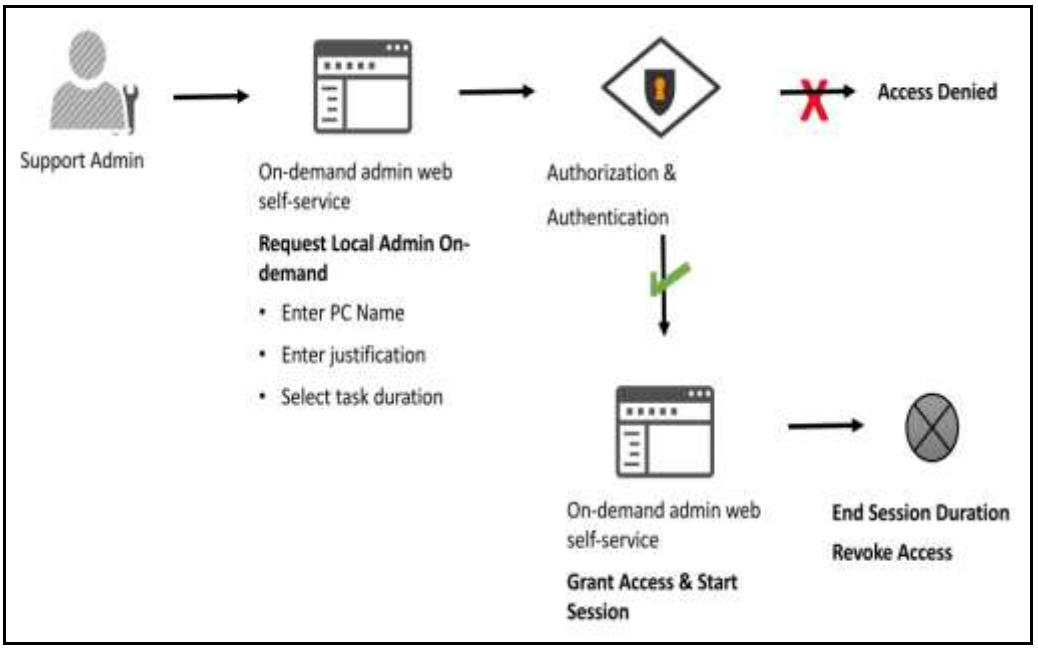

Figure 1. Overall Process of the Proposed Solution

8. Design Phase

The process is designed to be aligned with the previously mentioned system requirements to achieve all the functionalities in most automated, organized, and secured way. The on-demand desktop local admin right solution is designed to give the user limited time in administration access to protect the user's account without affecting or limiting its access capabilities. The system will make sure that admins will not get full privileged account. Hence, every time entitled users are required to do certain tasks on a remote desktop, they will get full administrative permission for specific time. After that, the permission will expire and the users will not be able to use their account to access any machine. This step will provide more security by making sure that any attack to malicious user and stealing user's credentials will minimize the risk of stealing the access or any privileges associated with the user account.

The solution is designed by preparing four stages. First, the privileged accounts will be prepared by maintaining a list of who is allowed to get a full administrative user account. The next stage is the protective stage in which the authentication and authorization requirements will be "set-up". The third stage is the operation stage in which the approved users will be granted access within specific time duration and they will have all privileges and access permissions to finish their required tasks. Once the time duration ends, the access will be revoked by removing the user account from the privileged admin group. The last stage focuses on monitoring the whole process. This include auditing and generating logs, reports, and alerts to observe the access requests for all privileged accounts. The history of all records will be kept in order to know who did what and at what time in case if any malicious attack is suspected. Checking such logs will also help in identifying any inside attackers who try to break any organizational rules.

8.1. Hardware and Software Requirements

The system will not require any hardware requirements other than the desktop computers that the end users are using. Since the system is tested and implemented at KOC, it was convenient to utilize all the available software that the company already owns to implement the whole system with zero cost. Moreover, we focused on using a unified technology to avoid any integration problems between products with different systems. The required software along with the purpose of each one is listed as the following:

1. Microsoft Desktop Operating System Windows 7 or 10: OS needed for end users' desktop computers.
2. Microsoft SharePoint Servers: develop a self-service Web interface to provide on-demand user access.
3. Microsoft Active Directory Domain Services: verify and authenticate users.
4. Microsoft SQL Server: provide real-time data logging database to track all user activities in the system.
5. Microsoft SQL Server Reporting Services: generate logs and reports based on the database created in the SQL Server to monitor the data.
6. Microsoft System Center Orchestrator: automate and integrate all the previous software components together to provide a workflow for the required solution.

9. High-Level System Architecture

The main system's components are demonstrated in the high-level architecture in Figure 2 bellow. Unlike regular architectures, the main components of this system are not linked together in lines. This is due the fact that Microsoft System Center Orchestrator automates and integrates all the deployed recourse in the environment. It will simply execute the created workflow in which it follows the order of execution. Therefore, instead of having a link between the self-service Web interface to the AD component and then another line back to the interface itself, System Center Orchestrator will manage the process and it will be able to know when to start the Web page, when to establish the authentication, and when to get back to the Web page. That's why the System Center Orchestrator groups the whole components as shown in Figure 2. The SQL Database is linked to each component in the system since it's used to keep record of everything to monitor the system. Therefore, this architecture can be considered as a high-level diagram for the solution while the workflow can give a more detailed picture of how each component is working to provide its required functionality.

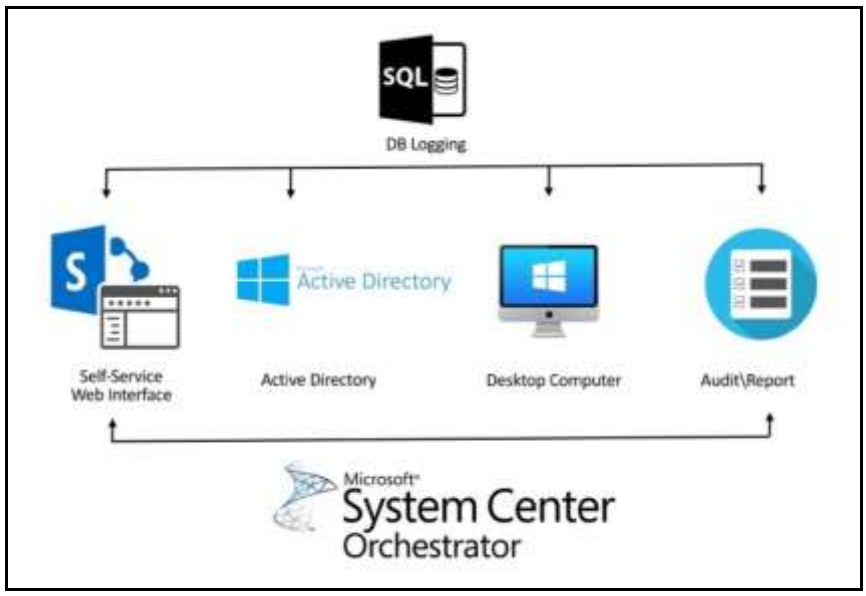

Figure 2. High-Level Architecture

10. Implementation Phase

The implementation of the proposed solution is mainly based on creating a workflow on Microsoft System Center Orchestrator to integrate and automate the main previously mentioned components. The orchestrator is like an engine that controls the whole process. It knows exactly when a certain component needs to be called. The workflow is shown in Figure 3. The next section explains the implemented workflow in detail.

10.1. Workflow

The Orchestrator will be running as a background process, and it will monitor any new request. The request will be initiated by the end user by using the Web portal interface, shown in Figure 4. That was developed using the SharePoint Server on the intranet. Next, the requester information will be collected to be linked with the created request. In each step, the orchestrator will keep a log of everything by writing records in the SQL database server. In addition, the status of each operation

will be checked to notify the user, send email, created error message, *etc.* Once the information of the user is collected, the user will be checked for authentication and verification using Active Directory services to check if the user is allowed to use the

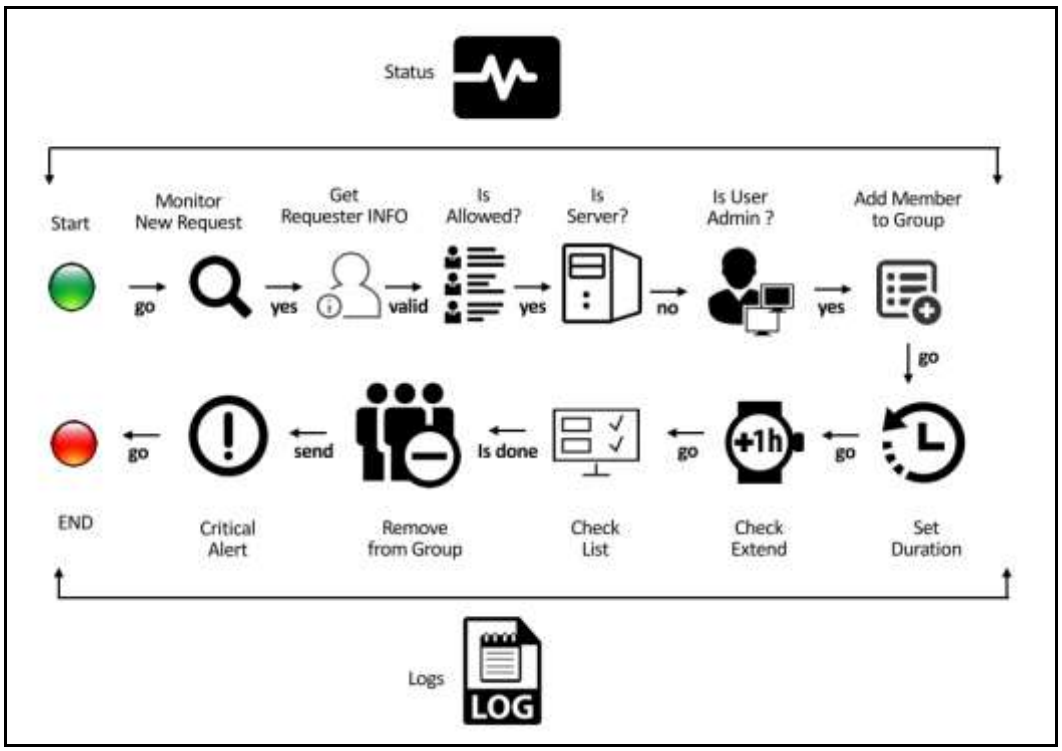

Figure 3. Workflow of System Center Orchestrator

system or not. If the verification and authentication check failed, the user will not be able to continue and an error message will be appeared. Otherwise, if the user is verified and authenticated successfully, the system will check the computer name that the user is trying to access to make sure it's not a server. If it turns to be a server, access will be denied for the user and an error message will be generated. If this was not the case, the system will do a third security check to make sure that this user is allowed to have a privileged account as an admin. This is performed by comparing the user name and ID with an approved list of all users who are approved to be admins based on their job role and responsibilities. This list is managed and approved by the Information Security Team to make sure that only monitored and specified users are the ones who are eligible to have a high privileged account according to the certain security standard. If the user is allowed to have admin rights, the process will begin and the user name will be added to the group of admin users. Further, an email with a welcome message to notify the user will be sent. It will include the duration of time that the admin user will be allowed to remote access the required computer. Once the access is granted, the user can finish the required tasks and responsibilities with a full administrative account privileged. The system will monitor the process and the duration of time the user is accessing the remote computer. Up to this point, the System Center Orchestrator is managing the communication process between SharePoint, Active Directory, and the desktop computer.

The System Center Orchestrator will hold-on the duration and keep checking if the duration is ended. At the same time, the system will check if the duration is extended. If so, the time will be increased and the user will be granted access for more time. If the duration of time was about to finish, an email will be sent to notify

the user that the duration will be finish after 15 minutes as shown in Figure 5. This step will allow the admin to know whether the remaining time is enough to finish the required tasks and responsibilities or no. If the required task is done before the end of duration, the admin can revoke the access on the system. Otherwise, if more time is needed, a request to extend the time shall be created. This will require justification to make sure that the request is extended on purpose. Once the time is done, the list of admins will

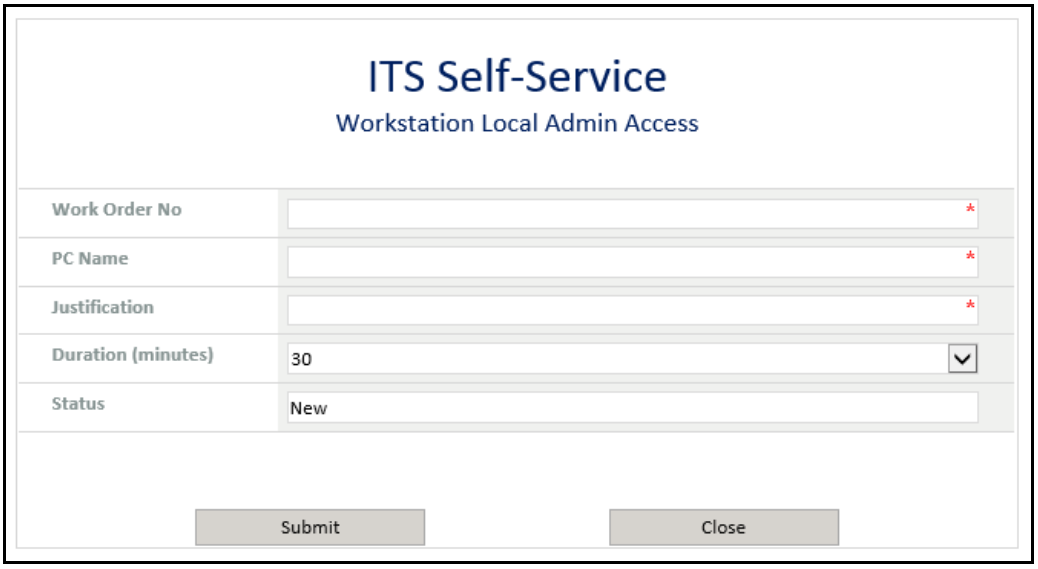

Figure 4. User Web Interface

be checked again and the user will be removed from the admin group be revoking the access. An alert will be sent to notify the user and the process will be end. Therefore, the access will only be granted once it's really needed, and as soon as the admin finish the tasks, the access will be revoked to make sure that this user account has no longer privileged access.

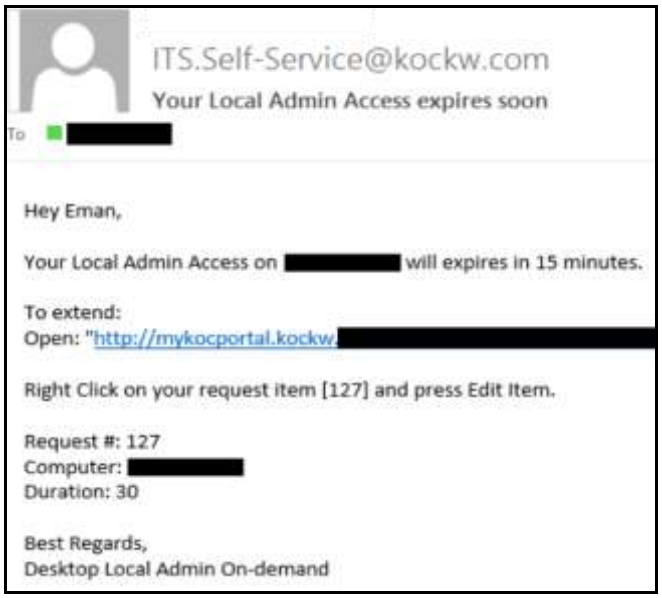

Figure 5. Sample Notification Email

10.2. Generating Logs, Reports, and Live Dashboard

Since the Orchestrator keeps a record of all the transactions done in the system and writes all the data using the SQL server, the reports can be generated to know the overall status of the system. A reporting center is implemented using the Web self-service. This is completed with the help of SQL Server Reporting Services. This functional requirement is needed to monitor all the generated requests for the purpose of keeping track of how each request is being accessed and what is the duration of time an administrator maintains full privileged account. In case of any attack incident, it's essential to know who has access on a certain PC to monitor the logs for any suspicious activities. Hence, this step is very critical not to only protect the organization from external attacks, but also internally by controlling the admin account's access. An audit report can be generated from the Web portal and it provides information such as: request ID, action time of the created request, work order number, requester name and user name, the computer name in which the requester wants to access remotely, the justification for this access, and the duration of access time. In addition, the report also demonstrates the status of the access to check whether the access is denied, granted, already completed, or revoked. In case of any failure message or additional details, the report lists the important system massages. For example, if an admin tries to access a server instead of a PC by providing the server name in the Web self-service, the access will be immediately denied and a message will be shown such as: "un-authorized access request, trying to access a server."

In addition to the logs and report, the system also provides a live dashboard to give an overall picture for the system. The dashboard is powerful when it comes to the comparison process. For example, the dashboard can display the total number of accesses in each month next to each other to provide an easy way view if one wants to know which month was in high demand for admin access. In addition, the dashboard is developed to show some important performance measures of the system such as the average processing time in each month. The interactive-live dashboard is shown in Figure 6, where some data is hidden for confidentiality. For example, before implementing the proposed solution, the user used to wait for days until an approval is required. This is due the fact that the process of requiring a full admin privileged access was based on manual work and human interactions. However, after automating the whole process, the average processing time in a month is only 21 seconds as shown in the dashboard figure below. The dashboard can be customized to show additional details and it's interactive, meaning that data will change based on user's selection, and based on that, it will reflect real time data.

Figure 6. The Live-Interactive Implemented Dashboard

11. Conclusion and Future Work

With the increased attacking activities toward different critical organization, controlling and monitoring the process of granting full local admin rights access became very vital. This is very important to lower the risk of hacking any user account that has full admin rights on the environment. This paper proposed and implemented a solution to develop an automated and managed process of requesting on-demand full administrative access for desktop computers. This solution is tested and implemented at Kuwait Oil Company. A Web self-service interface is developed to serve the users anytime to requests admin privileged access. A security check is implemented in two phases: authenticate the user with KOC active directory, and verify that the user has approval to obtain admin account. Once the verification and authorization is completed, the user will be able to start the session and remote access the requested computer to finish the required tasks within a specified time period. If the user finishes the required work before the end of the time duration, a request to revoke the access can be done by the user. If this was not the case, the system also allows the user to extend his/her access time with providing a valid justification. Once the duration of time is ended, the system will automatically revoke the admin access right from the user and the account will not be associated with any high privileged access. To monitor the status of this process, logs and repots can be generated from the system itself. Further, a live and interactive dashboard is developed to give an overview of the system and to demonstrate important performance measures.

As for future work, this system can be expanded to be implemented to monitor remote access on servers, not only computers. However, since the servers in any organization usually contain highly sensitive data, the process must contain extra security layers and close activity monitoring to know how the privileged accounts are using this service. This will positively impact the organization by providing a

secure process to observer and governor the use of high privileged accounts. Therefore, instead of having unlimited admin privileged associated to a user account to access a computer or server, this will be limited to specified time duration access and an automated-monitored process to provide more security and lower any security vulnerabilities in the system.

Acknowledgments

We would like to thank Mr. Mazen Ahmed, Senior Premier Field Engineer from Microsoft, for his dedication, hard efforts, and endless help in implementing this system. We would also like to appreciate the Portal Unit in KOC and Mrs. Eman Mousa's work, Service Desk Support, who helped in the coordination and testing phase. Finally, we would like to acknowledge the IT Services Team in KOC: Mr. Yacoub Al-Bash, ITS Team Leader, and Ms. Ghada Barak, TPL Specialist, for their support and encouragement to take this project to the next level from a proposed solution to an implemented real system.

References

[1] B. Wire, "Ttripwire study: Energy sector sees dramatic rise in successful cyber attacks", [Online]. Available: http://www.businesswire.com/news/home/20160407005104/en/Tripwire-Study-Energy-Sector-Sees-Dramatic-Rise.
[2] J. Pagliery, "The inside story of the biggest hack in history", [online]. Available: http://money.cnn.com/2015/08/05/technology/aramco-hack/.
[3] ISO.org. Iso/iec 27000 family - information security management systems. [Online]. Available: https://www.iso.org/isoiec-2700-information-security.html.
[4] "Microsoft. Privileged access management for active directory domain services", [Online]. Available: https://docs.microsoft.com/en-us/microsoft-identity-manager/pam/privileged-identity-management-for-active-directory-domain-services.
[5] "Gartner. Market guide for privileged access management", [Online]. Available: http://itsecurityleaders.com/wp-content/uploads/2015/09/BeyondTrust-Privileged-Account-Management-Research-Solutions-2015-Gartner-Market-Guide.pdf.
[6] D. Ferraiolo, D. R. Kuhn and R. Chandramouli, "Role-based access control", Artech House, 2003.
[7] A. Kern, "Advanced features for enterprise-wide role-based access control," in Computer Security Applications Conference, 2002. Proceedings. 18th Annual. IEEE, **(2002)**, pp. 333–342.
[8] D.R.Kuhn, E.J.Coyne and T.R.Weil, "Adding attributes to role-based access control", Computer, vol. 43, no. 6, **(2010)**, pp. 79–81.
[9] J. S. Park, R. Sandhu and G.-J. Ahn, "Role-based access control on the web", ACM Transactions on Information and System Security (TISSEC), vol. 4, no. 1, **(2001)**, pp. 37–71.
[10] R. Sandhu, D. Ferraiolo and R. Kuhn, "The NIST model for role-based access control: towards a unified standard", in ACM workshop on Role- based access control, vol. 2000, **(2000)**, pp. 1–11.
[11] M. Benantar, "Access control systems: security, identity management and trust models", Springer Science & Business Media, **(2006)**.

