Computer and Information Science; Vol. 15, No. 4; 2022
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Security and Performance through Operating System Services; Development of an Anti-Hacking System

Sahar Badri[1], & Daniyal Alghazzawi[1]

[1] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Correspondence: Daniyal Alghazzawi, Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia.

Received: July 13, 2022 Accepted: September 14, 2022 Online Published: September 23, 2022

doi:10.5539/cis.v15n4p66 URL: https://doi.org/10.5539/cis.v15n4p66

Abstract

An operating system is intricate software that is simple to compromise. Operating systems control a sizable number of services and updates, which, if not used properly, might provide problems. In this study, an anti-hacking system was created to guard against attacks on the operating system and boost its performance and security. Due to its increasing popularity among non-professional users, Microsoft Windows 8 was chosen in this context. The proposed solution may provide a significant improvement as software and has identified 21 and 33 services for security and operating system performance, respectively. Additionally, each service was individually detailed so that customers could quickly comprehend why they were needed and how to use them.

Keywords: anti-hacking, operating system, performance, security, services

1. Introduction

1.1 Introduce the Problem

Due to the increase in patches for computer infrastructure's performance, security, and functionality, the utilization of operating system services has gotten worse. Conventional techniques are necessary to address such patches by either rebooting the computer or restarting system services in order to go over this increasing threshold (Jo, Nam, & Shin, 2018). If it is successful, operating system performance will be improved and downtime will follow. Contrarily, there are times when delaying the update is not a good idea while implementing a security patch. The cost of unplanned downtime has forced system administrators and users to manage the increased susceptibility of a security error (Brookes & Taylor, 2016).

1.2 Explore Importance of the Problem

For the purpose of offering a special update functionality, an operating system is a dynamic environment with specific restrictions that must be handled. To overcome these difficulties in the implementation of the dynamic update mechanism for K42, Baumann et al. (2005) created an object-oriented research operating system. Furthermore, these issues have been addressed by Baumann et al. (2004) using a modern operating system function called hot-swapping. Due to their ability to provide a reliable mechanism for nearly all current technologies, multicore systems are now commonplace in all spheres of contemporary life. For many applications, it's crucial to have both great performance and strong security. Enhancing operating system performance is particularly crucial when considering speed and security concerns. Typically, performance declines when security increases, and vice versa (Samal & Dalai, 2018). Given the highly developed current state systems, there is no incremental improvement to the state-of-the-art to increase security without increasing labor.

1.3 Describe Relevant Scholarship

An operating system's main goals are to provide an appropriate environment for processor execution, safeguard processes from one another, and enable multiplex programme to access hardware. The most advanced operating system architecture is the result of numerous seminal and related research projects that achieve all functions and handle security issues. (Karnouskos & Kerschbaum, 2017; Gavrilova, et al., 2020; O'Regan, 2018; Schriner, 2018). Each application embeds a virtual address space in extant systems in which the required context is managed by the kernel. The virtual address space for each process carries its code for executing the functions to manipulate hardware. Virtual address spaces are shared by the kernel for protecting procedures from one another

between operating security systems as it limits access to the shared kernel functionality via hardware privilege levels for protecting procedures from one another (Votipka, et al., 2018).

In this study, Windows 8 as part of the Windows NT family of operating systems was challenged that deteriorated the extant state-of-the-art framework of multicore operating system design for increasing both security and performance instead of addressing one at a time.

Finding susceptibility successfully requires experience, which varies greatly between hackers and testers. Additionally, this study explains the functionality that is suggested for this programme and will assist customers in downloading any new Windows 8 updates from the internet. Depending on the category they choose, users can disable either manual or automatic service. As a result, this study creates an application that consumers can download several times.

1.4 State Hypotheses and Their Correspondence to Research Design

This study is significant in multiple ways. Firstly, the fundamental requirements and challenges for comprehensive operating systems' performance and security were identified. Furthermore, this study indicates that, due to different attack models and constrained environments, several services in current solutions are not available. Secondly, the system designed addresses all the challenges associated with and enhances current performance and security services, specifically in terms of hackers attack. Moreover, the application of the proposed design was based on a comprehensive prototype system for presenting real-world applications.

2. Related Work

Operating systems security is the process of safeguarding, OS availability, confidentiality, and integrity. OS security encompasses specific measures employed for protecting the OS from remote hacker intrusions, malware, worms, viruses, and threats (Jaeger, 2008). OS security involves all preventive-control practices that safeguard all computer assets which can be deleted, edited or stolen when the OS security is compromised. It allows various programs and applications to perform necessary tasks and prevent unauthorized interference. The proposed project will discuss the security of different operating systems and how it affects the overall security of web-based services and applications.

Because of the critical role which the operating system in the functioning of all computer systems, the operating system security or security absence will have fundamental effects on the overall computer system's security, containing all applications' security that run within the system. Compromising the base operating system might definitely expose the threat to all applications which run in the system (Bosworth & Kabay, 2002). Similarly, a lack of appropriate control and inhibition of the performance of distinct applications in the operating systems could result in an attack between different applications. For secure applications, the OS should be secure, and application compromises in the operating system should be handled. There are several risks that could result from compromising an application as a result of inappropriate operating systems security (Bosworth & Kabay, 2002). Therefore, the security of the operating systems is fundamental for the functionality of the computer systems, applications and programs. Malicious traffic is elevating in computer operating systems concerning both diversity and volume of attacks. Novel and additional complicated attacks were being utilized by hackers for exploiting network resources (Shafi, et al., 2010) Hackers are dependent upon some automated techniques. Hackers are competent enough to produce smart codes for making systems behave their way (Hyun, et al., 2017). This condition encourages the requirement of a system that can improve the security level of networks substantially. There can be several possible solutions for the issue, but intrusion prevention systems are the best of the breed because of several reasons (SDas & Nene, 2017).

They are quite able to detect and prevent diverse types of intrusion efforts and can be additionally enhanced in their accuracy and performance. Network resources are wide open for intruders regardless of prevention and detection systems (Wang, 2017). Conventional security measures cannot manage the situation adequately and complete intrusion prevention cannot be fulfilled. Intrusion prevention is an overall process to identify and prevent malicious contents throughout the network traffic going into or out of a network (JMiranda, et al., 2017). It is understood that a network-based intrusion prevention system positions the gateway to the network, identifies the malicious contents, and intercepts the network traffic throughout traffic, and takes rapid steps for stopping the attack efforts.

Intrusion efforts are done by hackers for exploiting the resources of firms and making massive advantages (Meng, et al., 2017). Organizations are investing a hefty amount of financial and other resources for protecting their information sources and network systems. At the same time, several intrusion attempts are elevating with the speed more than ever before. The secrecy of information is compromised through a successful intrusion

attempt made by any hacker to any organization as well as causing financial loss (Yerur et al., 2017). People will no longer prefer to do business with an organization that cannot secure its information resources and network systems. IT organizations use different technologies for assuring the protection of their worthy information resources, which include intrusion prevention systems, firewalls, anti-virus programs, and proxy servers (Fayaz, 2017).

In recent years, intrusion prevention systems being better as compared to other satisfying technologies have achieved the trust of professionals even though no technology offers the answers to all security-related questions (Sharafaldin, et al., 2018). Network resources are defended by intrusion prevention systems through their proactive approach. Attack signatures are used by intrusion prevention systems for identifying known attacks as well as identifying malicious behavior data contents being entered the network. Network protocol analyses are also performed by intrusion prevention systems for protecting against protocol violations (Bijone, 2016).The performance of intrusion prevention systems diminishes even though they strengthen the security level of a network substantially if offering huge data rates and high bandwidth. The majority of intrusion prevention systems target flow reconstruction and aggregation of transmission control protocol streams, but they are incompetent for handling gigabit rate links (Zitta, et al., 2018).

A nurse informaticist must be included in the process of switching information documentation systems. Informatics is a specialization that merges analytical science information management and handles data in the transitioning operating system, as stated by Yen et al. (2017). Adopting a successful development and implementation system is crucial when developing a new documentation system. The planning, analysis, design, implementation, and post-implementation support based on the SDLC are the main points of emphasis in this position description when it comes to directing the users' involvement in the system implementation.

Additional system resources are demanded because of elevated data rates for analyzing it, which include computability, memory, and bandwidth. The capability to handle other services is reduced due to the availability of such resources to intrusion prevention systems from hosts (Olufowobi, et al., 2019). A particular design is required for intrusion prevention systems to handle high data rates. It should be noted that intruders are always aware of this situation, and are well-equipped with resources for circumventing conventional intrusion prevention systems (Tanimu, et al., 2018). They improve the traffic from different systems to hacked systems for keeping it busy to handle massive data sets as well as launching additional sophisticated attacks that are complicated for defending and detecting. Intruders usually benefit from deceiving to hide their identities. This makes it complicated for stopping the attack.

Additionally, intrusion detection was modeled by Patcha and Park (2004) in mobile ad-hoc networks by adapting the signaling game in the multi-stage dynamic non-cooperation frameworks. The interactions between malicious attackers and the intrusion detection systems of their targets were modelled by Alpcan and Basar (2006) using a stochastic game. Alpcan and Basar (2006) have captured the operation using a finite-state Markov chain and naïve Q-learning for the intrusion-detection sensor system for finding the best strategies. Nguyen et al. (2009) have implemented the fictitious play game framework for modeling the relationship between defenders and hackers as a non-zero game sequence. The interactive behavior between the defender and hacker was explored by Lin et al. (2009) as information warfare and established a tree diagram based on game theory. These researchers have merely emphasized the classifications of regular and malicious users and evaluated the defensive mechanisms of systems despite the complicated games used in previous literature. They did not undertake the massive number of hackers nor the diversity in their intrusion behaviors.

3. The architecture of Operating Systems Services Based on Ontology Web Language

Planning is a crucial step in the development and implementation of a successful system. Planning enables users to evaluate and ascertain the system's requirements, flaws, and strengths, helping the system's execution. According to Mcgonigle and Mastrian (2021), planning determines the system's utility, costs, and resource requirements. Users are crucial to planning since they are aware of the requirements for documentation. Saba and McCormick (2015) anticipated that it is the role of the user to point out the needs as they related to documentation, the importance of the systems, the costs, and the system’s utility index. A user should be versed with information on the integration of the new system on the existing software and hardware and its implications, operating systems, and the organization’s documentation bulk.

Online services are a reliable platform for hosted services that reduces the need for deploying and managing on premise operating system services. OWL is hence ideal for managing operating system services, such as internet services. Important Exokernel services such as searching, securing, backing up, and creating a profile can be summed up semantically. Network services are best described as file sharing.

Ontology-based improves operating system management profile and search service. OWL enhances a web search engine in particular domains, like as geosciences teaching and research, where Noesis has utilised a constrained LEAD ontology. It makes use of domain ontology to give the user context so they can narrow down their search term and look for other resources that might be of particular interest to them. The idea behind ontology-based semantic personalized search is that by incorporating semantics into the system's information content, it may enhance the functionality of the search engines. Ontology is used to define the meaning of terms that appear in data files; these definitions can then be used to draw conclusions and obtain further information about the items of interest. Ontology-based approaches can be used to infer file relationships and to describe domains. The objective of classification in this situation is to assign each file to a class from a preset set of classes. The user context derives interest scores to current concepts in domain ontology by allocating implicitly in the personalized search which involves modeling. Each ontological user profile is fundamentally an instance of the reference ontology. Each concept is annotated by the user with an interest score, which has a basic value of one.

The ontological user profile is updated and the annotations for current concepts are altered by spreading activation as the user interacts with the system by viewing or choosing new documents. Therefore, the user context is maintained and updated significantly based on the ongoing behavior of a user. Moreover, back-up service is improved by ontology web language by utilizing Matcher ontology characteristic and CIM conceptual indexer. The CIM comprises a knowledge base that covers semantically enriched service data generated by service availability data from the service crawler, the automatic annotator, and user feedback gathered from the service finder.

Operating system security service further was deteriorated by addressing it using OWL. Security services are categorized into two main categories, such as authorization services and antivirus services. Moreover, the ontology-based policy translation approach authorized service in networks that imitates the behavior of expert administrators regardless of their mistakes. Network operating systems are devised for client computers and offer services so the difference between stand-alone operating systems and network operating systems is not always clear. Network operating systems offer services, such as replication and sharing services. However, it was observed that networking operating systems services are authorized as a class of sharing service as a directory service.

3.1 System Development

System analysis is the next phase during which the team considers the functional requirements of the system, analysis of end-user's needs, and whether the new system can meet the expectations (Tsui, et al., 2018). The system informaticist must be able to analyze the system's capabilities to meet the users' documentation needs and communicate this to the team. The system informaticist should analyze and ensure that the system is not only cost-effective but even adheres to the user's mission and purpose. Based on the user's analysis, relevant changes can be made to meet the requirements of the systems.

Online services are based on authentic hosted services platforms for reducing the requirement to execute and maintain IT-associated services. Therefore, operating system services are managed through ontology web language (OWL) as online services. This paper has semantically summarized essential operating services for searching, securing, packing-up, and creating profile and network services.

The researcher has used three different resources for developing the system: (1) Internet, (2) books, and (3) conferences. Furthermore, Microsoft's books reviewed that discuss the services mentioned regarding Windows. Similarly, exploits and incident handling, SysAdmin, Network Security, Audit, and Hacking techniques were other sources used in the development of a system. These sources were beneficial in determining the susceptibility of services in Windows that prevents hackers from attacking the computers (Figure 1).

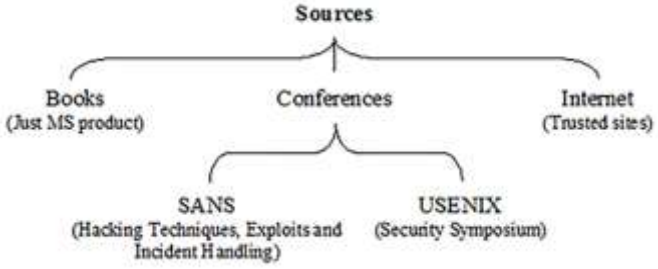

Figure 1. Sources used

3.2 Windows 8 Description

Once the assessment of the system has been completed, the next phase of designing the new system sets in which according to Tsui et al. (2018) describes the necessary specifications, features, and operations of the system as they relate to its functional requirements. A user must have all the competencies required to determine the components required for the system for it to perform effectively. In clarifying these requirements, prototypes or mockups of the system, reports and processes must be generated to help determine and eliminate glitches in the system that may prevent its operation (McGonigle & Mastrian, 2021). A user must demonstrate competency in ensuring flow sheets are functional and user-friendly for other nurses and healthcare professionals.

The operating system will be based on the following features: (1) technical descriptions; (2) added services; (3) performance features; and (4) security measures. The users were guided regarding the use of added programs in Windows 8, specifically for using one of the services embedded in the system. Similarly, the proposed system helps users in understanding the functions of different features such as a firewall. The system will additionally provide information regarding pop-up information related to the hackers' attack if any. Security ports are important features in this proposed system, which allow hackers to access the computer, if not appropriately controlled.

3.3 Categories

Following categories were added to this system

- Windows networking
- Disk and file management
- Remote administration
- Communication
- Clustering and load balancing
- Local service
- Collaboration
- Internet service, event monitoring, alerting, and logging
- System administration

3.4 Performance and Security

The performance and security of the operating system were measured through high performance and low-performance measurements. The system will perform higher when it introduces a new process in the operating system to take excessive memory. This aspect will consequently influence the machine's performance. Similarly, low performance will be measured when one or two services are shared in one processor links with multimedia applications such as music players. If the user shuts off the overall process by stopping all the services, it will improve performance.

In terms of security, the features will allow protecting the computer or opening the backdoor using a computer port. Moreover, it will be computer susceptible by allowing other individuals additional facilities to the computer. The prime objective was to consider the security and performance along with the notes for attracting the users at the time of reviewing the services. Therefore, users will have to review this service efficiently and appropriately, if they found any issues related to the performance or security of this service. At this instant, users can now assume about each service discussed previously, and whether each of them enhances the security or the performance of the windows manager.

4. Implementation

When the new design is deemed appropriate for the documentation in the organization, implementation plans are set in. In this phase, most of the system's codes are written, and data is moved from the old system to the new system (Tsui, et al., 2018). The phase involves taking the design and the system and putting its meaningful use. A user should seek professionals to learn about the application and use of the new system. Policies and procedures must be developed and incorporated into the new system during the implementation process to determine their effect on the new system. Coordinating the process of implementation and the inclusion of personnel is an important requirement for a user as part of the implementation team.

The service-oriented architecture paradigm followed the implementation of the developed approach. The Protégé 4 explanation engine was used for the implementation. The ontology of each operating system service was

created separately then the single components were implemented for generating the universal ontology. A new ontology search service was presented by developing ontology classes for files in the system, ontology-based query processing for the interface, and ontological user profile. The top-level classes were domain concept, content, and description in search services ontology.

Once implemented, the new system requires a continuous support practice that will ensure that the system is working well and issues arising due to the new system are directly addressed to ensure its functionality. Even with the best planning, glitches, opportunities, and unseen issues with the system always arise making the post-implementation support important in addressing them. As has been indicated by Laureate Education (2018), when people say it is not working, the post-implementation support and specifically, the system operator involved in the implementation should seek ways of rapidly fixing the issues within the system.

4.1 Domain Concept

It is the root of all domain classes which illustrate the domain considered in the documents.

4.2 Content

It refers to the contents in the file text media or document.

4.3 Description

It refers to the files generated by a user. This semantic classification is obtained through Text Mining for extracting knowledge in the document and allocate each file to a class of a predefined set of classes.

4.4 Ontological User Profile

This class refers to the user interest because this information can be captured from files on his computer and the domain can be specified a user is interested in. The user profile and object profile are top-level profile ontology-based classes (Figure 2). In operating systems, a user profile includes user settings and user documents, such as software settings, accessories, and control panel settings (Figure 3). Moreover, it comprises the information regarding the user such as his name, password, address, career, age, and gender.

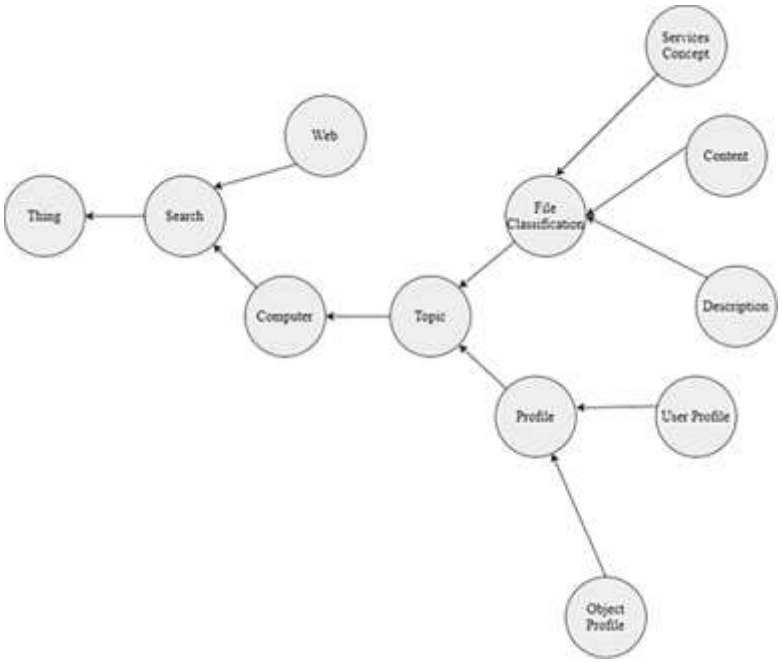

Figure 2. Search Service and Profile Ontology

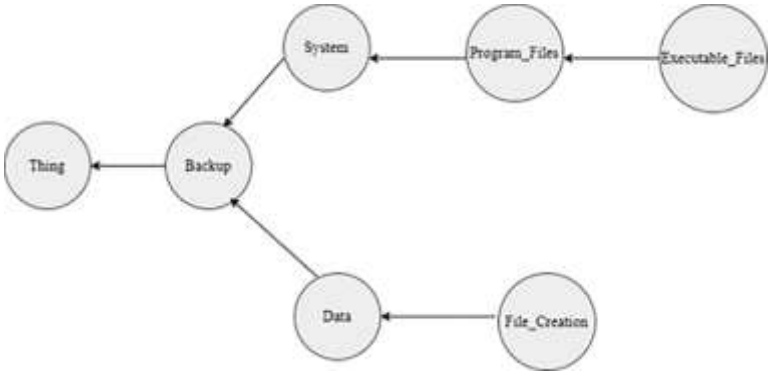

Figure 3. Backup Service Ontology

A subclass called context was also added comprising domain interest and web history. Information is stored in the operating system regarding any file generated on the computer such as its name, size, location, time, password, type, data, and attributes in the object profile.

The semantic personalized search service shows that user query can be executed alongside an ontology-based semantic classification for ontology classes and user profile, and return tuples of ontology values satisfying the query. This query was written in DL query, which is available in Protégé4 and an advanced query can be written for handling Pellet reasoner for providing complete and effective algorithms to address queries.

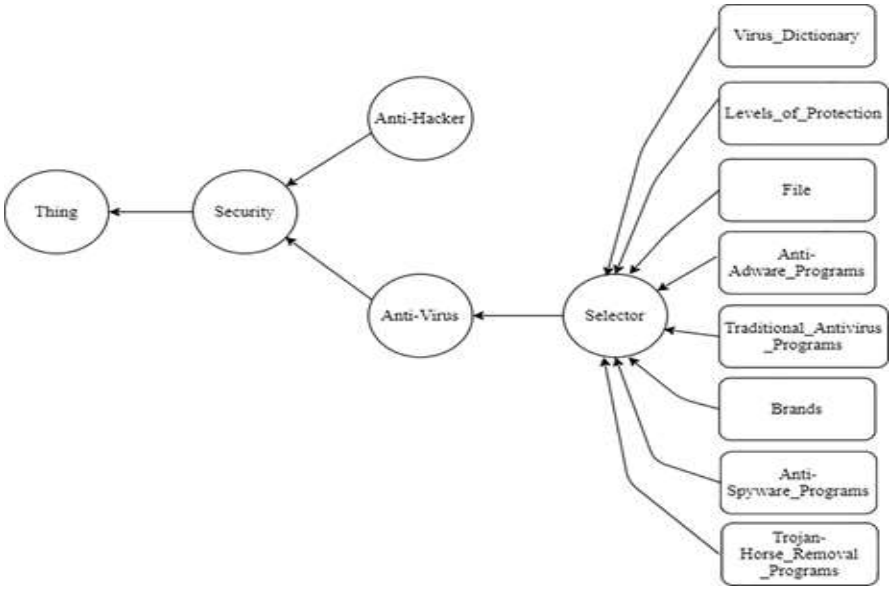

Figure 4. Security Service Ontology

Ontology-based query processing indicates how the ontology is generated and a query was received in natural language, and then transformed into an expression that can be declared to a description logic reasoner. Pellet reasoner complied with OWL formal semantics. The OWL ontologies were queried through SQWRL and therefore, can be utilized for retrieving knowledge inferred by SWRL rules. The fuzzy string-matching method was considered as a component of the proposed approach. It even assists in finding the closest individual names, which are missing in the queries entered by the user. Data and system program subclasses were the top-level ontology-based backup service.

The top-level structure of operating system services ontology is based on two major classes such as Exokernel and kernel service, where the kernel service is appropriate for matching with any operating system ontology. Lastly, the universal exokernel services ontology was generated by merging all the aforementioned services ontologies. The final universal services ontology was produced by using prompts.

MS-Windows 8 (Home Edition) was selected to implement the proposed system as this platform is used by the

majority of the non-administrator users for helping them in protecting their systems. Initially, all the services were checked and offered by MS-Windows 8 to execute the project. The following steps were taken to access these services:

- Click on Start ->
- Go to Settings -> Control Panel -> Administrative Tools -> Services.

Figure 5 shows the Services' Window.

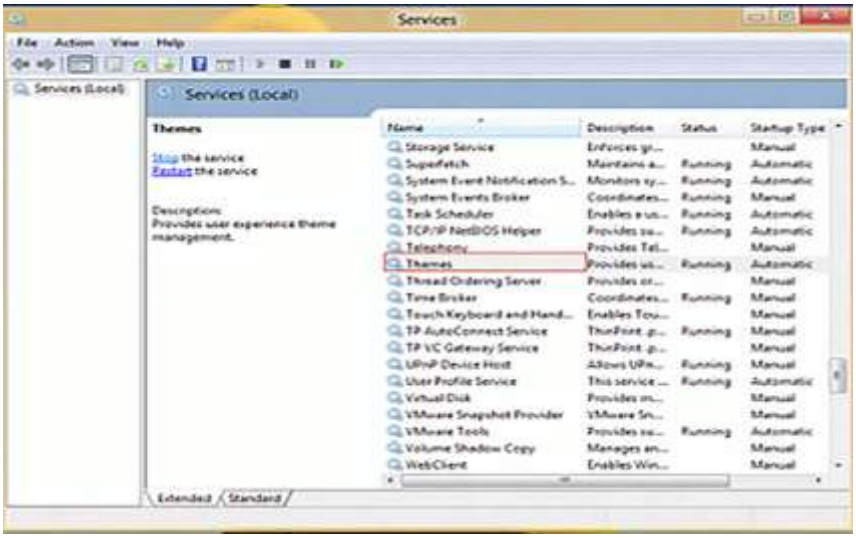

Figure 5. Services Window

- Click on Start -> Run.
- Then, type services.msc.
- This will bring up the Services' Window.

Click on Start then "Run" through the registry. Afterward, type Regedit with [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\UsbFlags]. By clicking this registry, users will be able to explore a list of all the services in the form of texts. There will be several services, which will help in increasing the performance and security of the windows system. The register is beneficial in creating a backup of the current setting before making any changes. Unfortunately, this feature allows hackers for using the Enable/Disable services on a remote machine. Therefore, this service was under consideration to protect the security of the windows system manager.

A list of services will be identified when searching the services section of the register. These services include Description, Start, and DisplayName. These services will be the same as found in the Services' Windows. A pseudo example is presented below:

- DisplayName: Alerter
- Description: Notifies selected users and computers of administrative alerts.
- Start: 0x4
- There are 5 options for Start:
- 0x0: Boot
- 0x1: System
- 0x2: Automatic
- 0x3: Manual
- 0x4: Disabled

In the execution phase, some of the services were running automatically by default, after reviewing all the services. Some ports in these services were opened in My Computer, which was never used. Hence, these services have reduced the performance and the security of My Computer, and there is a likelihood that these

services can create either denial of service (DOS) attacks or buffer overflow. In this regard, these steps were followed.

Search references (Conferences / Books / Online) for each service for understanding the objective of each of them. Classifying these services into 13 categories:

- Local Services
- Diagnostic Policy Service
- Security Center
- Print Spooler (Only disable this if you don't have a printer)
- Application Experience
- Secondary Logon
- Program Compatibility Assistant Service
- Portable Device Enumerator Service
- Offline Files
- Remote Registry
- Windows Error Reporting Service
- Windows Image Acquisition
- Windows Search
- Internet Services
- Distributed Link Tracking Client
- IP Helper
- Computer Browser
- Server (Disable this only if you are not connected to the Internet)
- TCP/IP NetBIOS Helper
- Windows Time
- Extract all the information about each service from the Operating System.
- Decide if this service is associated with specific ports.

There are additional services that should be considered when classifying the aforementioned services.

Performance: This service initiates a new process in the operating system or will take additional space from the memory. This service will influence the machine's performance. This service will also allow users for sharing one or multiple other services in one or more programs such as Windows Media Player. This service will; however, impact the machine performance to a limited extent. If the overall process is stopped by closing all the associated services, it might improve the system performance.

Normal: This service will show no security issues if it does not take excessive space from the memory.

Following recommendations should be considered by the users after reviewing all the previous issues:

Modem: Modem should be used by users for making outside connections. Therefore, the computer should be protected by the user rather than thinking about the performance.

Alone: If a user is not worried about the security of the system in case not using any connection including DSL, Cable, or Modem, then the core focus will be on the performance only.

Part of Network: A Hub or Switch can be used for connecting with the system. Therefore, a person should be assigned for tackling the security concern. This service will additionally help the user for protecting his computer instead of the system's performance.

Each aforementioned service will provide the following information:

- Recommend setting
- Service name
- Service status

- Service start type
- Windows 8 description

5. Conclusion and Future Work

This program was launched to trace all the approaches used by hackers for attacking computers. Several anti-hackers programs were identified. The third-party software protects the computers from these programs. However, it was found that none of the programs explore any loophole in the operating system itself. At the same time, regardless of using external tools, professional hackers will make efforts for hacking. They took benefit from the tools present in the operating system to access the machine via backdoors. Therefore, MS Windows 8 was selected as it was mostly used by inexperienced users.

It was identified that the majority of the users were unaware of the information regarding services present in the operating system and thus presented 21 services that contribute essentially to the security of the operating system. Furthermore, 33 other services were mentioned that can improve the performance of the operating system. This study might be used by developers in mitigating the challenges and loopholes present in selected operating systems and to control security concerns. However, they should provide a very comfortable migration platform for replacing the existing operating systems or at least offer awareness of the use of such systems to unprofessional users.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgement

The author acknowledges all the associated personnel, who, in any reference contributed in the completion of this study.

Conflict of Interest

The authors declares no conflict of interest.

References

A. Baumann, J. A., D. Da Silva, O. K., & R. W. Wisniewski, (2004). *Improving operating system availability with dynamic updates.* In Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-Demand IT Infrastructure. 21-27.

Baumann, A., Heiser, G., Appavoo, J., Da Silva, D., Krieger, O., Wisniewski, R. W., & Kerr, J. (2005, April). Providing dynamic update in an operating system. In *USENIX Annual Technical Conference, General Track* (pp. 279-291). https://doi.org/10.1145/1095810.1118622

Bijone, M. (2016). A survey on secure network: intrusion detection & prevention approaches. *American Journal of Information Systems*, *4*(3), 69-88.

Bosworth, S., & Kabay, M. E. (Eds.). (2002). *Computer security handbook*. John Wiley & Sons. https://doi.org/10.1002/9781118820650

Fayaz, H. (2017). Cloud Security Enhancement Through Intrusion Detection System. *International Journal of Advanced Research in Computer Science*, *8*(2).

Gavrilova, T. A., Globa, L. S., Golovko, V. A., Grabust, P. S., Guliakina, N. A., Kuznetsov, O. P., ... & Sharipbay, A. A. VV Golenkov–Editor-in-chief. https://doi.org/10.1007/978-3-030-60447-9_1

Jaeger, T. (2008). Operating system security. *Synthesis Lectures on Information Security, Privacy and Trust*, *1*(1), 1-218. https://doi.org/10.2200/s00126ed1v01y200808spt001

Jo, H., Nam, J., & Shin, S. (2018). Nosarmor: Building a secure network operating system. *Security and Communication Networks, 2018*. https://doi.org/10.1155/2018/9178425

Karnouskos, S., & Kerschbaum, F. (2017). Privacy and integrity considerations in hyperconnected autonomous vehicles. *Proceedings of the IEEE*, *106*(1), 160-170. https://doi.org/10.1109/jproc.2017.2725339

Laureate Education (Producer). Managing Health Information Technology, Baltimore, MD: Author. (2018). Retrieved from https://www.sec.gov/Archives/edgar/data/912766/000104746915009421/a2226925zs-4.htm

McGonigle, D., & Mastrian, K. (2021). *Nursing informatics and the foundation of knowledge*. Jones & Bartlett Publishers.

Meng, W., Li, W., & Kwok, L. F. (2017). Towards effective trust-based packet filtering in collaborative network environments. *IEEE Transactions on Network and Service Management*, *14*(1), 233-245. https://doi.org/10.1109/tnsm.2017.2664893

Nguyen, B. T., Oberberger, M. M., Parrott, G., & Wolf, B. D. (2009). *U.S. Patent No. 7,515,718*. Washington, DC: U.S. Patent and Trademark Office.

O'Regan, G. (2018). Overview of Operating Systems. In *World of Computing* (pp. 203-215). Springer, Cham. https://doi.org/10.1007/978-3-319-75844-2_10

Saba, V. K., & McCormick, K. A. (2015). *Essentials of nursing informatics*. McGraw Hill Education.

Samal, N., & Dalai, P. A. (2018). Performance survey of operating systems in iot environment. *Int. J. Comput. Sci. Mob. Appl*, *6*, 1-6.

Schriner, J. (2018). *The Legacy of Multics and Secure Operating Systems Today.*

Shafi, M. I., Akram, M., Hayat, S., & Sohail, I. (2010). Effectiveness of intrusion prevention systems (ips) in fast networks. *ArXiv preprint arXiv:1006.4546*.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, *1*, 108-116. https://doi.org/10.5220/0006639801080116

Tanimu, K. M., Pang, W., & Coghill, G. M. (2018). A Conceptual Framework of Starlings Swarm Intelligence Intrusion Prevention for Software Defined Networks. In *SICSA ReaLX*.

Votipka, D., Stevens, R., Redmiles, E., Hu, J., & Mazurek, M. (2018, May). Hackers vs. testers: A comparison of software vulnerability discovery processes. In *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 374-391). IEEE. https://doi.org/10.1109/sp.2018.00003

Wang, L. (2017). Big Data in intrusion detection systems and intrusion prevention systems. *Journal of Computer Networks*, *4*(1), 48-55.https://doi.org/10.12691/jcn-4-1-5

Yen, P. Y., Phillips, A., Kennedy, M. K., & Collins, S. (2017). Nursing informatics competency assessment for the nurse leader: instrument refinement, validation, and psychometric analysis. *JONA: The Journal of Nursing Administration*, *47*(5), 271-277. https://doi.org/10.1097/nna.0000000000000478

Yerur, S. V., Natarajan, P., & Rangaswamy, T. R. (2017). Proactive hybrid intrusion prevention system for mobile adhoc networks. *International Journal of Intelligent Engineering and Systems*, *10*(6), 273-283. https://doi.org/10.22266/ijies2017.1231.29

Zitta, T., Neruda, M., Vojtech, L., Matejkova, M., Jehlicka, M., Hach, L., & Moravec, J. (2018, December). Penetration testing of intrusion detection and prevention system in low-performance embedded IoT device. In *2018 18th International Conference on Mechatronics-Mechatronika (ME)* (pp. 1-5). IEEE. https://doi.org/10.23919/elmar.2018.8534645

Patcha, A., & Park, J. M. (2004, June). A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. In *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.* (pp. 280-284). IEEE. https://doi.org/10.1109/iaw.2004.1437828

Alpcan, T., & Basar, T. (2006, July). An intrusion detection game with limited observations. In *12th Int. Symp. On Dynamic Games and Applications, Sophia Antipolis, France* (Vol. 26). https://doi.org/10.1109/cdc.2003.1273013

Lin, J. C., Chen, J. M., Chen, C. C., & Chien, Y. S. (2009, July). A game theoretic approach to decision and analysis in strategies of attack and defense. In *2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement* (pp. 75-81). IEEE. https://doi.org/10.1109/ssiri.2009.27

S Brookes, S., & Taylor, S. (2016, September). Rethinking operating system design: Asymmetric multiprocessing for security and performance. In *Proceedings of the 2016 New Security Paradigms Workshop* (pp. 68-79). https://doi.org/10.1145/3011883.3011886

SDas, S., & Nene, M. J. (2017, March). A survey on types of machine learning techniques in intrusion prevention systems. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2296-2299). IEEE. https://doi.org/10.1109/wispnet.2017.8300169

JMiranda, J. M., Mtz, F. A. P., & Mata, M. A. M. (2017, June). Next generation systems—Scope and application of intrusion detection and prevention systems (IDPS) a systematic literature review. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-7). IEEE.1-7.

https://doi.org/10.23919/cisti.2017.7975925

Hyun, D., Kim, J., Hong, D., & Jeong, J. P. (2017, October). SDN-based network security functions for effective DDoS attack mitigation. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 834-839). IEEE. https://doi.org/10.1109/ictc.2017.8190794

Tsui, R., Davis, D., & Sahlin, J. (2018, July). Digital Engineering Models of Complex Systems using Model-Based Systems Engineering (MBSE) from Enterprise Architecture (EA) to Systems of Systems (SOS) Architectures & Systems Development Life Cycle (SDLC). In *INCOSE International Symposium* (Vol. 28, No. 1, pp. 760-776). https://doi.org/10.1002/j.2334-5837.2018.00514.x

Olufowobi, H., Hounsinou, S., & Bloom, G. (2019, November). Controller area network intrusion prevention system leveraging fault recovery. In *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy* (pp. 63-73). https://doi.org/10.1145/3338499.3357360

Appendix

Table 1. Recommendations List

Name	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Manual	Local System
Application Experience	Manual (Trigger Start)	Local System
Application Identity	Manual (Trigger Start)	Local Service
Application Information	Manual	Local System
Application Layer Gateway Service	Manual	Local Service
Background Intelligent Transfer Service	Automatic (Delayed Start)	Local System
Background Tasks Infrastructure Service	Automatic	Local System
Base Filtering Engine	Automatic	Local Service
BitLocker Drive Encryption Service	Manual (Trigger Start)	Local System
Block Level Backup Engine Service	Manual	Local System
Bluetooth Support Service	Manual (Trigger Start)	Local Service
Certificate Propagation	Manual	Local System
CNG Key Isolation	Manual (Trigger Start)	Local System
COM+ Event System	Automatic	Local Service
COM+ System Application	Manual	Local System
Computer Browser	Manual (Trigger Start)	Local System
Credential Manager	Manual	Local System
Cryptographic Services	Automatic	Network Service
DCOM Server Process Launcher	Automatic	Local System
Device Association Service	Automatic (Trigger Start)	Local System
Device Install Service	Manual (Trigger Start)	Local System
Device Setup Manager	Manual (Trigger Start)	Local System
DHCP Client	Automatic	Local Service
Diagnostic Policy Service	Automatic	Local Service
Diagnostic Service Host	Manual	Local Service
Diagnostic System Host	Manual	Local System
Distributed Link Tracking Client	Automatic	Local System
Distributed Transaction Coordinator	Manual	Network Service
DNS Client	Automatic (Trigger Start)	Network Service
Encrypting File System (EFS)	Manual (Trigger Start)	Local System
Extensible Authentication Protocol	Manual	Local System
Family Safety	Manual	Local Service
Fax	Manual	Network Service
File History Service	Manual (Trigger Start)	Local System
Function Discovery Provider Host	Manual	Local Service
Function Discovery Resource Publication	Manual	Local Service
Group Policy Client	Automatic (Trigger Start)	Local System
Health Key and Certificate Management	Manual	Local System
HomeGroup Listener	Manual	Local System
HomeGroup Provider	Manual (Trigger Start)	Local Service
Human Interface Device Access	Manual (Trigger Start)	Local System
Hyper-V Data Exchange Service	Manual (Trigger Start)	Local System
Hyper-V Guest Shutdown Service	Manual (Trigger Start)	Local System
Hyper-V Heartbeat Service	Manual (Trigger Start)	Local System
Hyper-V Remote Desktop Virtualization Service	Manual (Trigger Start)	Local System

Hyper-V Time Synchronization Service	Manual (Trigger Start)	Local Service
Hyper-V Volume Shadow Copy Requestor	Manual (Trigger Start)	Local System
IKE and AuthIP IPsec Keying Modules	Manual (Trigger Start)	Local System
Interactive Services Detection	Manual	Local System
Internet Connection Sharing (ICS)	Disabled	Local System
IP Helper	Automatic	Local System
IPsec Policy Agent	Manual (Trigger Start)	Network Service
KtmRm for Distributed Transaction Coordinator	Manual (Trigger Start)	Network Service
Link-Layer Topology Discovery Mapper	Manual	Local Service
Local Session Manager	Automatic	Local System
Microsoft Account Sign-in Assistant	Manual (Trigger Start)	Local System
Microsoft iSCSI Initiator Service	Manual	Local System
Microsoft Software Shadow Copy Provider	Manual	Local System
Multimedia Class Scheduler	Automatic	Local System
Net.Tcp Port Sharing Service	Disabled	Local Service
Netlogon	Manual	Local System
Network Access Protection Agent	Manual	Network Service
Network Connected Devices Auto-Setup	Manual (Trigger Start)	Local Service
Network Connections	Manual	Local System
Network Connectivity Assistant	Manual (Trigger Start)	Local System
Network List Service	Manual	Local Service
Network Location Awareness	Automatic	Network Service
Network Store Interface Service	Automatic	Local Service
Optimize drives	Manual	Local System
Peer Name Resolution Protocol	Manual	Local Service
Peer Networking Grouping	Manual	Local Service
Peer Networking Identity Manager	Manual	Local Service
Performance Logs & Alerts	Manual	Local Service
Plug and Play	Manual	Local System
PNRP Machine Name Publication Service	Manual	Local Service
Portable Device Enumerator Service	Manual (Trigger Start)	Local System
Power	Automatic	Local System
Print Spooler	Automatic	Local System
Printer Extensions and Notifications	Manual	Local System
Problem Reports and Solutions Control Panel Support	Manual	Local System
Program Compatibility Assistant Service	Manual	Local System
Quality Windows Audio Video Experience	Manual	Local Service
Remote Access Auto Connection Manager	Manual	Local System
Remote Access Connection Manager	Manual	Local System
Remote Desktop Configuration	Manual	Local System
Remote Desktop Services	Manual	Network Service
Remote Desktop Services UserMode Port Redirector	Manual	Local System
Remote Procedure Call (RPC)	Automatic	Network Service
Remote Procedure Call (RPC) Locator	Manual	Network Service
Remote Registry	Disabled	Local Service
Routing and Remote Access	Disabled	Local System
RPC Endpoint Mapper	Automatic	Network Service
Secondary Logon	Manual	Local System
Secure Socket Tunneling Protocol Service	Manual	Local Service
Security Accounts Manager	Automatic	Local System
Security Center	Automatic (Delayed Start)	Local Service
Sensor Monitoring Service	Manual (Trigger Start)	Local Service
Server	Automatic	Local System
Shell Hardware Detection	Automatic	Local System
Smart Card	Disabled	Local Service
Smart Card Removal Policy	Manual	Local System
SNMP Trap	Manual	Local Service
Software Protection	Automatic (Delayed Start, Trigger Start)	Network Service
Spot Verifier	Manual (Trigger Start)	Local System
SSDP Discovery	Manual	Local Service
Still Image Acquisition Events	Manual	Local System
Storage Service	Manual (Trigger Start)	Local System
Superfetch	Automatic	Local System
System Event Notification Service	Automatic	Local System

System Events Broker	Manual (Trigger Start)	Local System
Task Scheduler	Automatic	Local System
TCP/IP NetBIOS Helper	Automatic (Trigger Start)	Local Service
Telephony	Manual	Network Service
Themes	Automatic	Local System
Thread Ordering Server	Manual	Local Service
Time Broker	Manual (Trigger Start)	Local Service
Touch Keyboard and Handwriting Panel Service	Manual (Trigger Start)	Local System
UPnP Device Host	Manual	Local Service
User Profile Service	Automatic	Local System
Virtual Disk	Manual	Local System
Volume Shadow Copy	Manual	Local System
WebClient	Manual (Trigger Start)	Local Service
Windows All-User Install Agent	Manual (Trigger Start)	Local System
Windows Audio	Automatic	Local Service
Windows Audio Endpoint Builder	Automatic	Local System
Windows Backup	Manual	Local System
Windows Biometric Service	Manual	Local System
Windows Color System	Manual	Local Service
Windows Connect Now – Config Registrar	Manual	Local Service
Windows Connection Manager	Automatic (Trigger Start)	Local Service
Windows Defender Service	Automatic (Trigger Start)	Local System
Windows Driver Foundation – User-mode Driver Framework	Manual (Trigger Start)	Local System
Windows Error Reporting Service	Manual (Trigger Start)	Local System
Windows Event Collector	Manual	Network Service
Windows Event Log	Automatic	Local Service
Windows Firewall	Automatic	Local Service
Windows Font Cache Service	Automatic	Local Service
Windows Image Acquisition (WIA)	Manual	Local Service
Windows Installer	Manual	Local System
Windows Management Instrumentation	Automatic	Local System
Windows Media Player Network Sharing Service	Automatic (Delayed Start)	Network Service
Windows Modules Installer	Manual	Local System
Windows Remote Management (WS-Management)	Manual	Network Service
Windows Search	Automatic (Delayed Start)	Local System
Windows Store Service (WSService)	Manual (Trigger Start)	Local Service
Windows Time	Manual (Trigger Start)	Local Service
Windows Update	Manual (Trigger Start)	Local System
WinHTTP Web Proxy Auto-Discovery Service	Manual	Local Service
Wired AutoConfig	Manual	Local System
WLAN AutoConfig	Manual	Local System
WMI Performance Adapter	Manual	Local System
Workstation	Automatic	Network Service
WWAN AutoConfig	Manual	Local Service