A Proposed Framework for Access Control in the Cloud and BYOD Environment

Khalid Almarhabi, Kamal Jambi, Fathy Eassa and Omar Batarfi,

Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

Summary

The new trend to bring your own devices (BYODs) to work to connect to the network is a fast-growing and popular trend. These devices represent a high security threat to the organization's network. BYODs can be contaminated with spyware and viruses that enable the device to access private information. BYODs can have disastrous results when improperly used. BYOD risks include unauthorized changes to policies and information, leaking sensitive information to the public, the financial and legal implications of a breach, and the loss of productivity for the organization. The intention of this paper is to introduce a new architectural framework to control the risks of BYODs. This solution is derived from large volumes of research into information privacy and security to manage and control access to enterprise networks by BYODs. The proposed architecture aims to reduce restrictions and enforce access control policies in the cloud and BYOD environment in a soft and secure manner with an independent platform.

Key words:

Bring Your Own Device, access control, policy, security

1. Introduction

BYOD refers to the new trend of company employees and executives using their own devices in the office for work and then taking the devices home at the end of the day [1-3]. When an organization allows BYODs as client devices for access to data and enterprise applications, the servers are usually on the cloud. One study estimated there will be more than one billion BYODs used in workplaces across the world in 2018 [4]. Another study observed that 95% of participants used their own devices to perform work functions [5]. These numbers are rising as BYODs provide the organization with many benefits. These benefits include boosting morale, productivity, employee satisfaction and job ownership and providing employees with greater work flexibility and mobility [6].

However, BYODs in the workplace create considerable challenges for organizations. There is a risk of poor organizational control over individual personal devices. One risk is staff accessing unauthorized areas of the enterprise system. Such issues as unauthorized use of cloud-based applications without adhering to company policies [6] are real challenges. These are referred to as 'shadow IT.' Other risks include employees being distracted by such social media platforms as Facebook and Twitter, which is contrary to company policies. This risk puts both individuals and the organizations they work for at constant risk from cyberattacks because of poor access controls [7]. Conversely, the control must consider user privacy and rights. While a number of organizations strike the right balance between controlling BYODs work and personal use, others' monitoring practices may push personal privacy boundaries. This possibility is why people using their personal devices as BYODS need to understand their rights [8]. It is a concern for employees that their employer can access their personal devices without permission under the pretext of management. The vast difference between using BYODs for personal and work use can create disagreements between the two parties regarding access control [9].

Malicious apps downloaded by employees can affect the corporate network and BYOD devices. "Keyloggers, malware, and cyber-attacks have greatly increased the potential for unauthorized access to, and information theft from, endpoints" [10]. Almost no organization or personal device is immune to attacks from malicious applications [10-12]. The risk increases when staff bypasses the system's limitations by rooting or jailbreaking devices to access areas that are off-limits. When these steps are taken, BYODs and the cloud network are opened to malicious attacks through transferring, processing, and storing data phases. Changing authorization policies is one of the main targets for attackers. These risks become even higher when an organization does not have permission from the owner of a BYOD to check for viruses, spyware, and malware before connecting to the organization's system. Mobile operating systems, such as Windows, Android, and IOS, are vulnerable to cyberattacks, as shown in (Table 1) [13].

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

Table 1: List of different types of attacks in different operating systems

Name	*Attack(s)*	*Mobile OS*
Zeus (Zitmo)	• Mobile Banking Attacks • TAC Thefts • Illegal Transactions	• Symbian • Win Mobile • BlackBerry • Android
DroidDream	• Theft of Private Data • Downloading Malicious Applications	• Android
Android.Bmaster (SmartRoot)	• Revenue Generation • Theft of Private Data	• Android
AnserverBot	• Theft of Private Data	• Android
Ikee.B	• Revenue Generation • Theft of Private Data	• iPhone
TigerBot	• Theft of Private Data • Changing Device Settings	• Android

Almost all operating systems have the potential for attack. Solutions must be compatible across different operating systems, which means that the software must be able to run on any hardware or software platform. Harmful malware can collect and leak sensitive data, track the user, and change organizational policies, as shown in (Fig. 1) [14].

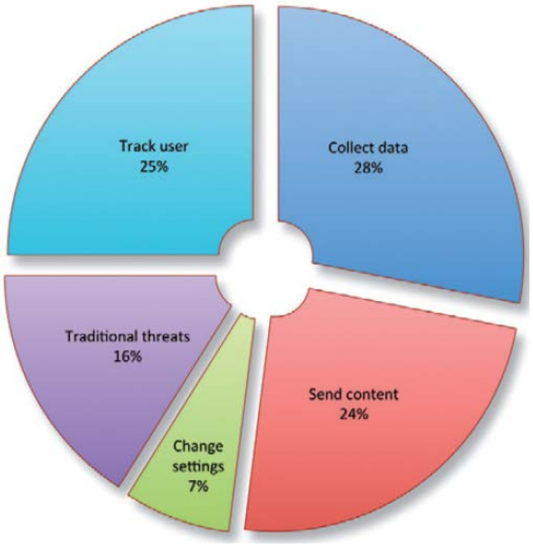


Fig. 1 What malwares do with BYOD devices [29]

When a BYOD is stolen or lost or an employee leaves, all of that sensitive data are placed at risk. More than 9 million smartphones are lost or stolen each year [15, 16], which is a considerable challenge for organizations. Something that most people do not know is that it is possible to retrieve data erased from handheld smart devices. Even data erased from the operating systems of devices can still be recovered by professionals [17, 18].

It is important for organizations to implement the right processes and procedures to minimize the risks. However, many organizations fail to have a security policy in place. The organizations that do often have policies that fail to address the technical or organizational requirements for information security [19]. With this failure, the control of personal devices is the biggest security risk for companies [20, 21]. There are a number of applications available for managing and controlling personal devices for greater security. Still, researchers observed that a number of organizations fail to control BYODs in an appropriate manner [7]. This is of concern to BYOD owners with 57% of study participants [22] expressing worry regarding unauthorized access to their devices by employers. The largest worry for organizations and their workers is the risk of unlawful access to enterprise systems due to security risks caused by the BYOD trend.

2. Related Work

We have investigated the most recent trends addressing the access control issues in BYODs concerning information security [23]. We have analyzed the essential and comprehensive requirements needed to develop an access control framework in the future. Four requirements are necessary to develop future solutions, as listed below.

2.1 Check BYOD Device Security

It is highly important to ensure the security of BYOD devices. These devices must meet all of an organization's requirements to avoid threats that may distort or destroy data in the mobile devices. The proposed solution should not restrict access to a specific device for each user that would go against the advantages of BYOD trends. In other words, device registration limits the use of BYODs. The proposed solution also needs to work in a way that does not conflict with user privacy and rights.

2.2 Enforce Access Control Policy

This enforcement refers to the organization's technical policy that must be followed. This policy includes the policy for BYOD devices to meet the minimum requirements of security, the authentication phase, and the authorization phase. Each user must have access to certain resources, which is set by the policy administrators. Mandatory access control is one of the best mechanisms to implement an access control policy. Restricted access control should not be based on a specific place or time such that employees can benefit from the full advantages of the BYOD trend.

2.3 Platform Independence

The proposed solution should implement solutions that are compatible with all BYOD operating systems. This solution will help reduce risks in these devices, regardless of the operating system, and ensure covering all different operating systems.

2.4 Secure Access Control Policy

It is useless to develop new techniques without protecting them. The access control policy can be easily modified by malicious actions from internal BYOD devices or external threats that attack dzata and policies. The protection must cover all phases of transfer, process, and storage in the BYOD and cloud environment. Several existing solutions focus on user data with less concern about cloud side attacks.

Existing approaches can be evaluated based on these distinguishing requirements. Table 2 shows this information.

Table 2: Previous approaches comparing to our proposed framework

Paper citation	Check BYOD Device Security	Enforce Access Control Policy	Platform Independence	Secure Access Control Policy
[24]	P	Y		
[25]		Y		P
[26]	Y	Y	Y	
[27]	P			P
[28]	Y	Y		P
[21]	P	Y	Y	
Our proposed framework	Y	Y	Y	Y

Y = yes

P = partly

However, we think that based on the literature review, each of the previous studies addresses a single issue and does not provide a complete solution to address access control issues. As a result, these solutions are still insufficient and warrant further research. We can integrate and develop several parts together as described in the next section. This paper focuses on the technical side. This study will not cover the developments of procedures and rights that need to be adhered to by users.

3. Proposed Framework

Cloud services are divided into three main models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). All of these models are managed by a cloud manager. We proposed a new security manager tool called Software as a Service (AaaS) for a public cloud provider. The new SaaS will be an available tool for any organization’s SaaS to communicate with it through a cloud manager to perform the security tasks regarding access control in BYODs and the cloud environment. We have considered several issues in the design of the framework and attempted to make it easy to add and use by limiting operating requirements and not affecting existing BYODs and cloud environments. Our proposed framework is based on a multi-agent system because it is independent software that runs on the behalf of a network user. With this BYOD environment, the software agent effectively works due to its adaptability, mobility, transparency, raggedness, and self-starts and stops. This environment can reduce the costs and the required resources during the coordination with other machines. The proposed framework can be divided to three parts: the client BYOD, the owner device, and the security manager (Figure 2). We will explain each software agent of the proposed framework in this paper.

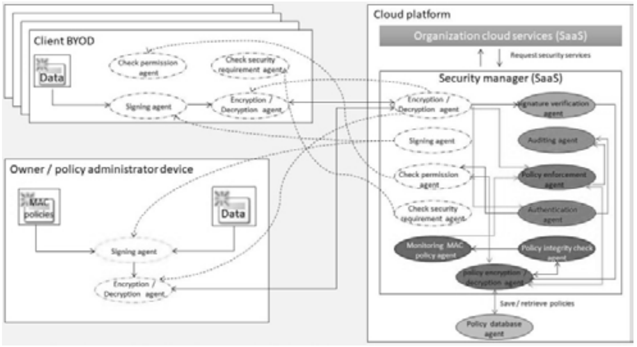

Client BYOD
Data
Check permission agent
Check security requirement agents
Signing agent
Encryption / Decryption agent
Owner / policy administrator device
MAC policies
Data
Signing agent
Encryption / Decryption agent
Cloud platform
Organization cloud services (SaaS)
Request security services
Security manager (SaaS)
Encryption / Decryption agent
Signing agent
Check permission agent
Check security requirement agent
Signature verification agent
Auditing agent
Policy enforcement agent
Authentication agent
Monitoring MAC policy agent
Policy integrity check agent
policy encryption / decryption agent
Save / retrieve policies
Policy database agent

Fig. 2 Our proposed framework for the BYOD and cloud environment

3.1 Owner / Policy Administrator Device

The first part is for the person who is responsible for setting the policy, whether it is the Chief Security Officer (CSO), the policy administrator or the organization’s owner. His device can be a BYOD or normal PC with a trusted operating system, such as Security-Enhanced Linux (SELinux). He/she can perform a critical main job, which is to set up the security classification level for the users and the initial data.

(i) MAC policy

MAC stands for the Mandatory Access Control policy. The MAC mechanism imposes harsh access limits that mostly cannot be bypassed unintentionally or intentionally. The MAC is effective, since it applies a clearance that is owned by every user. MAC establishes whether a user can have access to a certain file. This can be done flexibly using the JavaScript Object Notation language (JSON). There are four main security classification levels for both users (subjects) and resources (objects), which are top secret, secret, confidential, and unclassified. The policy administrator is responsible for determining the use and resource security classification levels as required in the MAC. The JSON file and data will be encrypted and signed after the data are digitally signed. Here is an example of a JSON file:

```
{
  "Version": "2018-1-17",
  "username": “John”,
    "compartmentalization": { “computer
  science”,
    "security classification level":
  "Secret",}
}
```

(ii) Data

This includes all resources that we want to upload and store in the cloud.

3.2 Security Manager

This is the core of our proposed framework that manages all of the components, as described below. The framework is located in the cloud side and works when it is called by a software as a service. This proposed framework achieves four main aspects using different agents: checking BYOD device security, enforcing the access control policy, working with independent platforms, and securing the access control policy.

(i) Controller agent

This static agent manages all other agents. It has the ability to create instances from mobile agents and send them to other different devices using their IP addresses. This agent also contains the Application Programming Interface (API) to communicate with other software as a service in the cloud.

(ii) Check security requirement agent

This agent is mobile and created by the controlling agent to travel to all connected devices using the organization’s software as a service (SaaS) in the cloud. This mobile agent checks if BYOD devices are trusted by whether they meet security policy requirements for an organization, such as updated antivirus software and installing an agent manager, fingerprints, and a VPN connection. In this research, we will take the requirement for an updated antivirus app for each device as an example. In general, this agent will provide a summary of what the user must do to use his / her device in accordance with the security requirements for an organization.

(iii) Authentication agent

When the devices meet the security policy requirements, the process of authentication is started as a basic requirement. This agent ensures that a user is valid for an account. Each user must have a unique identity, and this agent demonstrates their identity. Two different types of authentication are used to enhance the protection of the system.

(iv) Check permission agent

When the authentication agent finishes, this agent will take the username from the authentication agent and search for his/ her security classification level in the database. Next, the agent will send the username to the user’s BYOD device to make a preliminary decision on giving access. This agent implements the concept of MAC and uses the MAC policy that includes the security classification level to make the access decision. The purpose of this agent is to increase the performance by reducing wasted time if the user is not allowed legitimate access before the request goes through the internet to the cloud side. The agent also shows a user their permissions when they access specific resources. For example, a list of files will appear to the user with permission details next to each file, such as read only, read and write. If a user passes this step, the next check for permission will occur in the cloud side by the “Policy enforcement agent”.

(v) Signing and signature verification agents

These mobile agents ensure that a message was sent by a known user and was not modified in transit. These agents generate digital signatures for each JSON policy file and data issued by owner or BYOD users, as shown in (Figure 3). The signature verification agent in the security manager verifies the digital signature by comparing the decrypted hash value with the generated hash value of the original JSON policy and the initial data. If the values are equal, this means that this message has not been modified. This figure shows the process of verifying the digital signature.

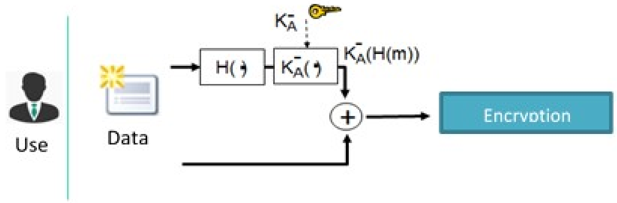


Fig. 3 Generating the digital signature in a BYOD device

(vi) Encryption and decryption agent

This mobile agent ensures that only authorized users and agents can access and read the data. This agent keeps the transmitted data secret. The agent has the ability to encrypt and decrypt all access control policies and data transmitted between the security manager in the cloud side and user devices. This agent initially uses an asymmetric algorithm (also known as public-key cryptography), only to exchange the symmetric key (which is the Advanced Encryption Standard (AES)) and use it for a while. Figure 4 shows the decryption process to obtain the AES key to decrypt the mac policy.

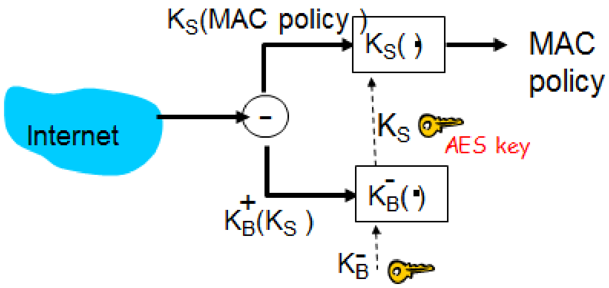


Fig. 4 Decryption of the MAC policy

(vii) Policy enforcement agent
This static agent is the second and main mechanism to enforce the access control policy to determine what users can access on the cloud. This agent enhances the access control mechanism and supports the DAC to provide a stronger access control mechanism. The agent implements the concept of Mandatory Access Control (MAC) using the Bell–LaPadula model, as in (Figure 5), to achieve confidentiality as required in our case study. This agent works with the allowed access from the "check permission agent" to ensure that the user has legitimate access, and the decisions have not been modified by an attack through the transfer phase.

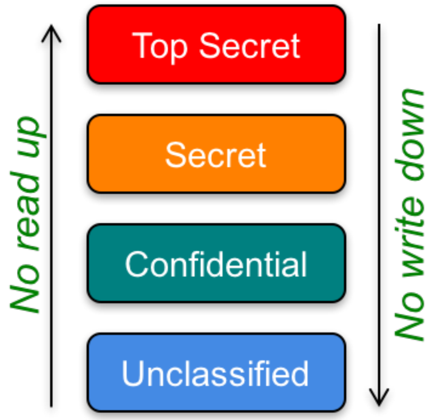


Fig. 5 Bell–LaPadula model

(viii) Policy monitoring and integrity check agents
These static agents save a copy from the first hash value of the MAC policy that was generated or updated by the owner. It will continuously use it to compare it with other new generated hash values of the same MAC policy. It should always be equal. The purpose of these agents is to ensure no modifications were made to the MAC policy by an attack during the processing phase. It will ensure that the MAC policy has been sent by the policy administrator only and has not been modified in the transit phase. This agent will inform the controller agent and policy administrator when an attack happens.

(ix) Auditing agent
This static agent records all successful and failed attempts to access the system. This agent also records all of the decisions taken by the "Policy enforcement agent" to grant or deny access to a user. These records include such information as the username, date, time, resources, and decision. Such information helps the policy administrator to monitor, analyze, conduct regulatory compliance, know the causes of the crime (in the case that it occurs), perform disaster recovery, and develop the system.

(x) Policy encryption and decryption agent
This static agent has the ability to encrypt and decrypt all data that are stored or retrieved from the access control database to protect the data in the storage and transfer stages. This agent uses the Advanced Encryption Standard (AES).

(xi) Policy database agent
This static agent is responsible for communicating with other databases as a service (DBAASs), database management systems (DBMSs) or distributed database management systems (DDBMSs). This agent will exchange the data and handle them between different software architecture styles and patterns.

3.3 Client BYOD Device

The client can use their own device after checking if BYOD devices are trusted by meeting the security policy requirements. The check security requirement agent does this job, and it will travel to their BYOD devices when they want to access the cloud. When the agent meets the requirements, other agents will travel to their device to perform its functions, as described above. These agents are the encryption and decryption agent, the check permission agent, and the signing agent. Clients can create and share data by suggesting security classification levels, and then the owner approves it. Clients do not need to work from a specific place and time or through a certain device.

All sequence diagrams have been drawn for this proposed framework after dividing it into 7 sub-frameworks based on the main tasks. The most important sequence diagrams will be explained, which are creating and modifying policies or data by policy administrators, monitoring the MAC policy in the security manager, and creating and modifying data by clients.

When the owner wants to create or modify existing policies or data through the user interface, the signing method is called the "signing agent", which adds a digital signature to the message, as shown in (Figure 6). After that, the data and policies will be encrypted by the "encryption and decryption agent" before going through the internet.

Another instance from the "encryption and decryption agent" in the cloud side will decrypt the message. The verification method is called the "signature verification agent", and it verifies the digital signature. When it is passed, the processing method in the "policy enforcement agent" will implement the MAC mechanism and deny all illegitimate request access. Saving methods will be called in both the "policy encryption and decryption agent" and the "policy database agent" to save the allowed policies and data. The ack methods in both agents will confirm the process of saving the data. Finally, the recording method in the "auditing agent" records all the details of the final decision made by the "policy enforcement agent".

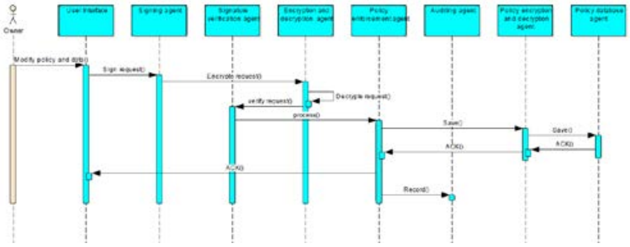


Fig. 6 Sequential diagram for creating and modifying policies or data by policy administrators

Monitoring the MAC policy in the security manager is the main task to protect policies during the process and storage phases, as shown in (Figure 7). The "controller agent" starts activating the "policy integrity check agent". It will call the request hash key method to get a copy of the policy from the database. Next, the reply hash key method will generate a hash value from these policies. After that step, this value will be sent to the "monitoring MAC policy agent" to be compared with the original one. If they are equal, the process will repeat continuously. Otherwise, the "monitoring MAC policy agent" will call the report error method in the "controller agent" to stop the authentication, record the issue, delete the existing policies, and inform the owner.

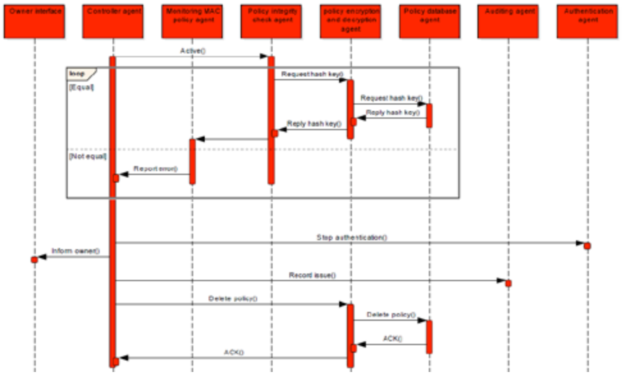


Fig. 7 Sequential diagram for monitoring the MAC policy in the security manager

When clients want to create or modify existing data through the user interface, the check permission method is called to make a preliminary decision on whether to give access based on the clearance and security classification levels. If a decision is allowed, the signature method is called in the "signing agent" to add the digital signature to the data, as shown in (Figure 8). After that, data will be encrypted by the "encryption and decryption agent" before going through the internet. Another instance agent from the "encryption and decryption agent" in the cloud side will decrypt the data. Next, the verification method is called in the "signature verification agent" to verify the digital signature. When it is passed, the processing method in the "policy enforcement agent" will implement the MAC mechanism and deny all illegitimate request access. The modified data methods will be called in both the "policy encryption and decryption agent" and the "policy database agent" to save the allowed data. The ack methods in both agents will confirm the process of saving the data. Finally, the recording method in the "auditing agent" records all the details of the final decision that made by the "policy enforcement agent" or the "check permission agent" in the preliminary decision.

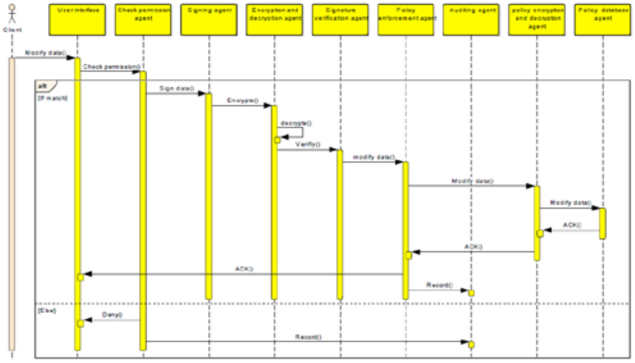


Fig. 8 Sequential diagram for creating and modifying data by clients

4. Implementation and Testing

Implementing and testing the proposed framework is required to verify and validate the solution. It is required to ensure that there is no fault, error or failure in the system. The implemented prototype has two core components. The first is the client and owner application, and the second is the security manager as Software as a Service (SaaS) in the cloud. Mobile agent software is required in these components. There are a variety of agent frameworks can be used, such as Concordia, Aglets, and Jade. In the client and owner BYOD devices, we built an application by using c# and Java in the Microsoft visual studio framework. The JavaScript Object Notation language (JSON) is used in these codes to implement the MAC. We use real BYOD devices based on the Windows operating system to install the app and connect to the cloud. In the security manager, we used the same above environments to build two software as services. One of them is our security manager, and the other one is the organizational software as a service that is connected to our security manager. These two software as services are deployed in the Google cloud platform and use its storage as a database.

Black and white box tests are used first to examine the functionality and structure of the proposed framework. The validation was completed successfully by validating some of the requirements that are used in our proposed framework. We used four cases to test the proposed framework based on potential attacks, as shown in the following (Figure 9).

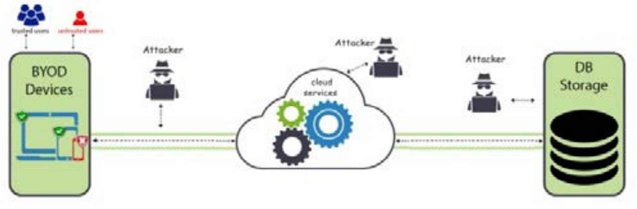


Fig. 9 Potential attacks that may occur in the cloud and BYOD environment

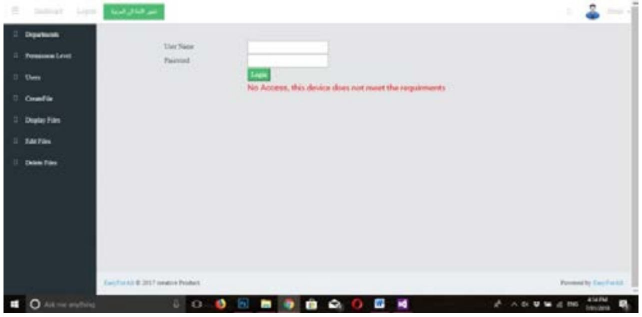
Case 1: The use of an untrusted device by trusted and untrusted users.

Case 2: The use of a trusted device with trusted users who want to access illegitimate resources.

Case 3: The access control policy is attacked during the process and storage phases.

Case 4: Test 20 access control policies during the transfer phase with the correct digital signature, the incorrect digital signature, the original cipher text, and the modified cipher text.

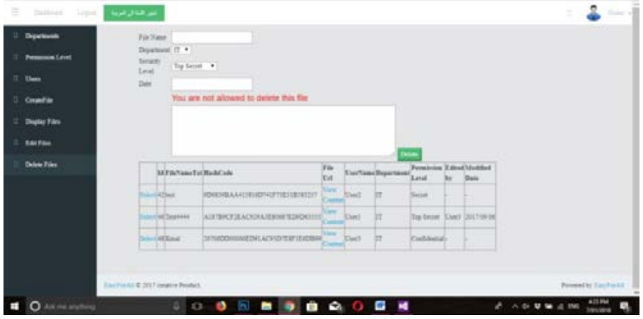
For the first case, the "check security requirement agent" was able to detect an untrusted device that does not meet the organization requirement of an updated antivirus program, as seen in (Figure 10). In this scenario, the application will not be allowed to connect to the cloud.



No Access, this device does not meet the requirements

Fig. 10 Interface showing the untrusted BYOD device

For case 2, the system detect users that want to access illegitimate resources by comparing the MAC security classification level of the user with the security classification level of the wanted resource using the Bell–LaPadula model, as in (Figure 11).



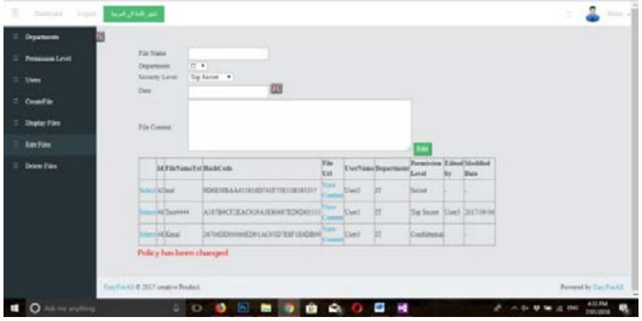
File Name
Department: IT
Security Level: Top Secret
Date

You are not allowed to delete this file

	Id	FileNameTxt	HashCode	File Txt	UserName	Department	Permission Level	Edited by	Modified Date
Select	42	test	[illegible]	View Content	User2	IT	Secret	-	-
Select	44	Test++++	[illegible]	View Content	User1	IT	Top Secret	User1	2017-09-06
Select	48	Kmal	[illegible]	View Content	User3	IT	Confidential	-	-

Fig. 11 Interface showing denied access to illegitimate resources

For case 3, the proposed framework faced a number of attacks that modified the access control policy during the process and storage phases. The hash value changed and was detected, as seen in (Figure 12).



File Name
Department: IT
Security Level: Top Secret
Date
File Content

	Id	FileNameTxt	HashCode	File Txt	UserName	Department	Permission Level	Edited by	Modified Date
Select	42	test	[illegible]	View Content	User2	IT	Secret	-	-
Select	44	Test++++	[illegible]	View Content	User1	IT	Top Secret	User1	2017-09-06
Select	48	Kmal	[illegible]	View Content	User3	IT	Confidential	-	-

Policy has been changed

Fig. 12 Interface showing the detection of the changed MAC policy

Finally, we test the 20 accesses of the control policy during the transfer phase with different characteristics. Five of them had the correct digital signatures, five of them had the incorrect digital signatures, five of them had the original cipher text, and five of them had the modified cipher text. Both the "encryption and decryption agent" and the "signature verification agent" detected all modified access control policies, as shown in (Figure 13).

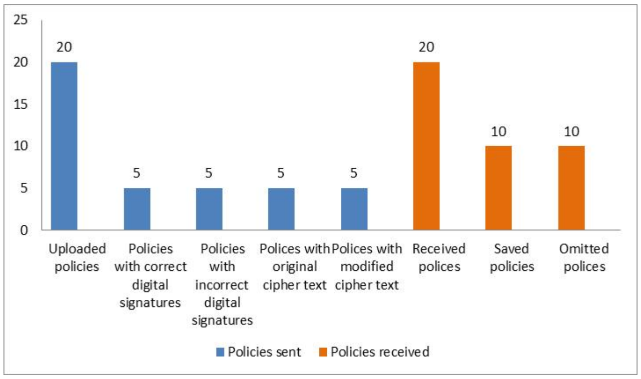


Fig. 13 Statistics shows the number of uploaded policies, received policies, saved policies, and rejected policies.

To sum up, the process of verifying and validating was completed successfully and we detected all attacks as we planned. The functionality and structure of the proposed framework was examined and gives positive feedback with no faults, errors or failures in the system.

5. Conclusions

In this paper, we introduce a solution to the access control issues in BYODs and the cloud environment. We aimed to design a solution that maintains the features of BYODs, such as mobility and improved flexibility. This solution is based on four main requirements, which are checking the BYOD device security, enforcing the access control policy, working with independent platforms, and securing the access control policy. We integrate all of these requirements and build our proposed framework based on the multi-agent system due to its adaptability, mobility, transparency, raggedness, and self-start and stops. Most other existing solutions solve specific issues without comprehensive consideration of the effects of these solutions on the BYOD environment or their users. We attempted to reduce the restrictions and increase the flexibility and mobility with a soft implementation of the policy. We also tried to protect user’s privacy by avoiding the use of Mobile Device Management (MDM) solutions. We have also built the first prototype of the system by implementing and testing the proposed framework in real environments. The outcome of verification and validation show excellent results and positive feedback. The future work will implement this proposed framework in real scenarios of exiting and running businesses. The collection of data through this implementation over the long-term will be analyzed to evaluate the system. Beta testing will also be employed to improve the total security of the proposed framework.

References

[1] Information Commissioner's Office (ICO), "Bring your own device," ed, pp. 1-14.
[2] T. Shumate and M. Ketel, "Bring your own device: benefits, risks and control techniques," in SOUTHEASTCON 2014, IEEE, 2014, pp. 1-6.
[3] A. V. Herrera, M. Ron, and C. Rabadão, "National cyber-security policies oriented to BYOD (bring your own device): Systematic review," in Information Systems and Technologies (CISTI), 2017 12th Iberian Conference on, 2017, pp. 1-4.
[4] M. Dhingra, "Legal issues in secure implementation of bring your own device (BYOD)," Procedia Computer Science, vol. 78, pp. 179-184, 2016.
[5] A. B. Garba, J. Armarego, D. Murray, and W. Kenworthy, "Review of the information security and privacy challenges in Bring Your Own Device (BYOD) environments," Journal of Information privacy and security, vol. 11, pp. 38-54, 2015.
[6] P. Beckett, "BYOD–popular and problematic," Network Security, vol. 2014, pp. 7-9, 2014.
[7] D. Dang-Pham and S. Pittayachawan, "Comparing intention to avoid malware across contexts in a BYOD-enabled Australian university: A Protection Motivation Theory approach," Computers & Security, vol. 48, pp. 281-297, 2015.
[8] M. M. Singh, C. W. Chan, and Z. Zulkefli, "Security and Privacy Risks Awareness for Bring Your Own Device (BYOD) Paradigm," INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, vol. 8, pp. 53-62, 2017.
[9] S. Blizzard, "Coming full circle: are there benefits to BYOD?," Computer Fraud & Security, vol. 2015, pp. 18-20, 2015.
[10] B. Morrow, "BYOD security challenges: control and protect your most sensitive data," Network Security, vol. 2012, pp. 5-8, 2012.
[11] A. V. R. Herrera, Mario and C. Rabadao, "National Cyber-security Policies oriented to BYOD (Bring Your Own Device): Systematic Review," IEEE 12th Iberian Conference on Information Systems and Technologies (CISTI), 2017.
[12] Checkpoint website. (2015). Security Report. Available: http://www.checkpoint.com/resources/2015securityreport/CheckPoint-2015-SecurityReport.pdf
[13] M. Eslahi, R. Salleh, and N. B. Anuar, "MoBots: A new generation of botnets on mobile devices and networks," in Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on, 2012, pp. 262-266.
[14] S. Enterprise, "Internet Security Threat Report 2014," ed, 2015.
[15] B. Yulianto and R. Layona, "An Implementation of Location Based Service (LBS) for Community Tracking," ComTech: Computer, Mathematics and Engineering Applications, vol. 8, pp. 69-75, 2017.
[16] PricewaterhouseCoopers (PWC), "The Global State of Information Security Survey," 2015.
[17] N. Zahadat, P. Blessner, T. Blackburn, and B. A. Olson, "BYOD security engineering: A framework and its analysis," Computers & Security, vol. 55, pp. 81-99, 2015.

[18] J. Girard, "Top Seven Failures in Mobile Device Security," Gartner, 2013.

[19] M. M. Ratchford, "BYOD: A Security Policy Evaluation Model," in Information Technology-New Generations, ed: Springer, 2018, pp. 215-220.

[20] J. Thielens, "Why APIs are central to a BYOD security strategy," Network Security, vol. 2013, pp. 5-6, 2013.

[21] P. de las Cuevas, A. Mora, J. J. Merelo, P. A. Castillo, P. Garcia-Sanchez, and A. Fernandez-Ares, "Corporate security solutions for BYOD: A novel user-centric and self-adaptive system," Computer Communications, vol. 68, pp. 83-95, 2015.

[22] H. Schulze, "BYOD & Mobile Security Report," 2014.

[23] K. Almarhabi, K. Jambi, F. Eassa, and O. Batarfi, "Survey on access control and management issues in cloud and BYOD environment," International Journal of Computer Science and Mobile Computing, vol. 6, pp. 44-54, 2017.

[24] G. Costantino, F. Martinelli, A. Saracino, and D. Sgandurra, "Towards enforcing on-the-fly policies in BYOD environments," in Information Assurance and Security (IAS), 2013 9th International Conference on, 2013, pp. 61-65.

[25] L. L. Bann, M. M. Singh, and A. Samsudin, "Trusted Security Policies for Tackling Advanced Persistent Threat via Spear Phishing in BYOD Environment," Procedia Computer Science, vol. 72, pp. 129-136, 2015.

[26] S. Chung, S. Chung, T. Escrig, Y. Bai, and B. Endicott-Popovsky, "2TAC: Distributed access control architecture for" Bring Your Own Device" security," in BioMedical Computing (BioMedCom), 2012 ASE/IEEE International Conference on, 2012, pp. 123-126.

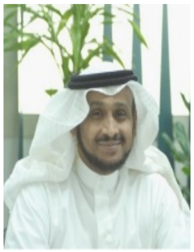
[27] K. AlHarthy and W. Shawkat, "Implement network security control solutions in BYOD environment," in Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on, 2013, pp. 7-11.

[28] U. Vignesh and S. Asha, "Modifying security policies towards BYOD," Procedia Computer Science, vol. 50, pp. 511-516, 2015.

Khalid Almarhabi is a Lecturer of computer science at Umm Alqura University in Makkah. He holds a BSc degree in computer science from King Abdulaziz University in Jeddah, Saudi Arabia. He also holds an MSc degree in Information Technology from Queensland University of Technology in Brisbane, Australia. He is currently pursuing a Ph.D degree at King Abdulaziz University. His research interests are secure BYODs, access control policies, information system management, cloud computing, and e-learning.

KAMAL M. JAMBI was born in Makkah, Saudi Arabia in 1960. He received his B.S. in computer science from the University of Petroleum and Minerals, Dhahran, Saudi Arabia in 1982, his M.S. degree in computer science from Michigan State, East Lansing, Michigan, USA in 1986 and his Ph.D degree in computer science from Illinois Institute of Technology, Chicago, IL in 1991. Prof. Jambi has been a professor in the Computer Science Department at King Abdulaziz University, Jeddah, Saudi Arabia since 2009. His areas of interest includes OCRs, Image processing, NLPs, and big data. He has also been the chairman of the CS department at FCIT and Vice Dean of Graduate Studies and Scientific Research. He was the PI for several projects funded by KACST.

Fathy E. Eassa received his B.Sc degree in electronics and electrical communication engineering from Cairo University, Egypt in 1978, his M. Sc. degree in computers and Systems engineering from Al Azhar University, Cairo, Egypt in 1984, and his Ph.D degree in computers and systems engineering from Al-Azhar University, Cairo, Egypt with joint supervision with the University of Colorado, U.S.A, in 1989. He is a full professor with the computer Science dept, Faculty of Computing and Information technology, King Abdulaziz University, Saudi Arabia. His research interests include agent based software engineering, cloud computing, software engineering, big data, distributed systems, and exascale system testing.



Omar Batarfi received his B.S. degree in Computer Science from King Abdulaziz University, Jeddah, Saudi Arabia in 1989 and his M.S. degree in Artificial Intelligence from George Washington University, Washington, D.C., USA in 1996. He received his Ph.D. from University of Newcastle Upon Tyne, UK in 2014.

From 2008 to 2016, he was an Assistant Professor with the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. He is currently an Associate Professor of Networking Security at Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include Big Data, Cloud Computing and Information Security.

Dr. Batarfi was a Visiting Scholar from 2014 to 2015 with the University of North Carolina, USA.