An Evaluation of the Proposed Framework for Access Control in the Cloud and BYOD Environment

Khalid Almarhabi[1], Kamal Jambi[2], Fathy Eassa[3], Omar Batarfi[4]
Department of Computer Science
King Abdulaziz University, KAU
Jeddah, Saudi Arabia

***Abstract*—As the bring your own device (BYOD) to work trend grows, so do the network security risks. This fast-growing trend has huge benefits for both employees and employers. With malware, spyware and other malicious downloads, tricking their way onto personal devices, organizations need to consider their information security policies. Malicious programs can download onto a personal device without a user even knowing. This can have disastrous results for both an organization and the personal device. When this happens, it risks BYODs making unauthorized changes to policies and leaking sensitive information into the public domain. A privacy breach can cause a domino effect with huge financial and legal implications, and loss of productivity for organizations. This is a difficult challenge. Organizations need to consider user privacy and rights together with protecting networks from attacks. This paper evaluates a new architectural framework to control the risks that challenge organizations and the use of BYODs. After analysis of large volumes of research, the previous studies addressed single issues. We integrated parts of these single solutions into a new framework to develop a complete solution for access control. With too many organizations failing to implement and enforce adequate security policies, the process needs to be simpler. This framework reduces system restrictions while enforcing access control policies for BYOD and cloud environments using an independent platform. Primary results of the study are positive with the framework reducing access control issues.**

***Keywords*—Bring your own device; access control; policy; security**

I. Introduction

Bring your own device (BYOD) is the trend where employees use personal handheld devices for work as well as for personal use [1-3]. Employees own the devices so they take them home each day. Organizations with cloud network systems usually allow the use of BYODs for accessing data and enterprise applications. This has huge advantages for both employees and employers. One study estimated there will be more than one billion BYODs used for work in 2018 [4]. Another study said 95% of participants used personal handheld devices to perform work functions [5]. More and more people are using BYODs because of the benefits. It boosts morale, productivity, employee satisfaction and job ownership as well as work flexibility and mobility [6].

This raises organizational challenges, in particular, device and network security. Using BYODs means organizations have poor control over them without adequate security policies. Organizations have concerns about unauthorized access to cloud-based applications that bypass company policies [6]. This is referred to as 'shadow IT' where activities take place on a company network without specific organizational approval. The use of BYODs also risks employees accessing social media during work hours contrary to company policy. BYODs in the workplace exposes companies to greater security risks; in particular, the heightened risk of cyber-attack as it is hard to control access out of hours [7].

This is a conundrum for organizations as they need to consider user privacy and rights along with protecting networks from attacks. Some organizations get the balance between controlling BYODs for work and personal use right. Others' monitoring practices can violate an employee's personal privacy and rights when using personal handheld devices for personal reasons. It is important BYOD users understand their rights [8]. It is possible for employers to access private information without permission under the guise of management practices without good security mechanisms in place. This will cause problems for employees and employers if the process for managing access control to enterprise applications after hours is not transparent [9].

Employees have the right to use personal devices in any manner they like as long as they do not breach company policies. Unknowingly they can download malware and malicious applications, which can have a negative effect on corporate networks as well as their own devices. 'Keyloggers, malware, and cyber-attacks have greatly increased the potential for unauthorized access to, and information theft from, endpoints' [10]. With most organizations and personal devices vulnerable [10-12], risks increase when staff bypass system limitations by rooting or jailbreaking devices to access off-limit areas. This threatens personal devices and the cloud network with a malicious attack when transferring, processing, and storing data.

Organizational risks escalate when there are no access policies. These policies need permission from owners to check devices for viruses, spyware, and malware before connecting to its system.

Windows, Android, and IOS mobile operating systems are all vulnerable to cyber-attack (Table I) [13]. Malware collects and leaks sensitive data, tracks users, and changes authorization policies (Fig. 1) [14], which means a high degree of vulnerability. No operating system is immune from attack and organizational solutions need to be compatible on all operating systems.

TABLE I. 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

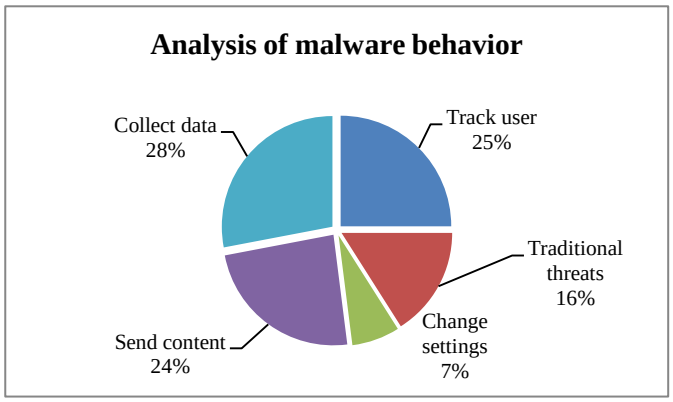

Fig. 1. What malwares do with BYOD devices [29]

Sensitive data is at risk when a personal device is lost or the employee leaves. With more than 9 million smartphones lost or stolen each year [15, 16], this is a considerable challenge. Even when data is deleted from a device and its operating systems, experts can retrieve that data [17, 18].

Many organizations fail to implement appropriate security policies for employees using BYODs. Where organizations do have security policies, they are inadequate because they do not address technical or organizational requirements for information security [19]. This makes controlling personal devices the biggest security risk for companies [20, 21]. Although there are applications available to manage and control personal devices, organizations are not using them in an appropriate way [7].

One study showed these concern BYOD owners when 57% of respondents [22] expressed worry about employers accessing personal devices without their authorization. By far the most concerning issue is the risk of unauthorized access to enterprise systems through BYODs.

In short, BYOD boosts morale, productivity, employee satisfaction and job ownership as well as work flexibility and mobility but it has some issues with employers such as poor controlling, violating an employee's personal privacy and rights, spreading malware and malicious applications, and lacking appropriate security policies for employees using BYODs.

II. RELATED WORK

We investigated the latest BYOD trends to address control systems to protect information security [23]. We analyzed the requirements for developing a suitable access control system and found there are four requirements.

A. Check BYOD Device Security

Any solution must meet an organization's security policies, while not breaching user privacy and rights. There has to be the ability to check the security levels installed on each individual device to avoid threats that can change or destroy data. The challenge is to find a solution that does not restrict user access either as it conflicts with the purpose of BYOD. Previous solutions call for device registration before use on a company network. Device registration limits the use of BYODs especially when a device is lost or replaced.

B. Enforce Access Control Policy

Mandatory access control is the best mechanism for protecting an organization from the risk of using BYODs. However, restricting access to certain locations or work hours negates the benefits of BYODs for both the employer and the employee. There needs to be minimum requirements for security, authentication, and authorization phases for BYODs to meet. Policy administrators need to set access controls to the resources each user requires. Organizations must then enforce all technical and access control policies.

C. Platform Independence

Any proposed solution should be compatible with all BYOD operating systems to reduce the risks from these devices to keep the process simple and flexible.

D. Secure Access Control Policy

Developing new policies is of no value without protecting them. Without protection, it risks malicious actions from BYODs that may have downloaded malware that modifies access an access control policy. There are also the risks from external threats that attack data and policies from BYODs. A secure access control policy must protect the process of transferring, processing, and storing data when a BYOD interacts with the cloud environment. There are several solutions that focus on user data without addressing possible side attacks on the cloud.

TABLE II. 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

Table II shows an evaluation of how previous approaches compare to our proposed protection framework.

From the literature review, we see previous studies address single issue without providing a complete access control solution. As a result, these solutions are insufficient and require further research. We integrate several parts of these to develop a new solution for BYOD access control. We describe this in Section III. This paper focuses on the technical side of the solution. It does not attempt to develop the required processes a user needs to follow to support an access control policy.

III. Proposed Framework

Cloud services have three main models managed by a cloud manager: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). We propose a new security manager tool called Software as a Service (AaaS) for public cloud providers. The AaaS framework gives any organization's SaaS the ability to use cloud manager to perform security checks before granting BYOD access to the cloud environment. We considered several issues when designing the framework. It was important to make the tool easy to add and use without affecting existing BYOD and cloud environments. We achieved this by limiting operating requirements.

We based the framework on a multi-agent system, because the software runs independently on behalf of a network user. This makes it adaptable, mobile, transparent, and it automatically starts and stops. This reduces the costs and the required resources when a BYOD interacts with other machines. The proposed framework is divided to three parts: the client BYOD, owner device, and the security manager (Fig. 2). Each software agent is explained in this paper.

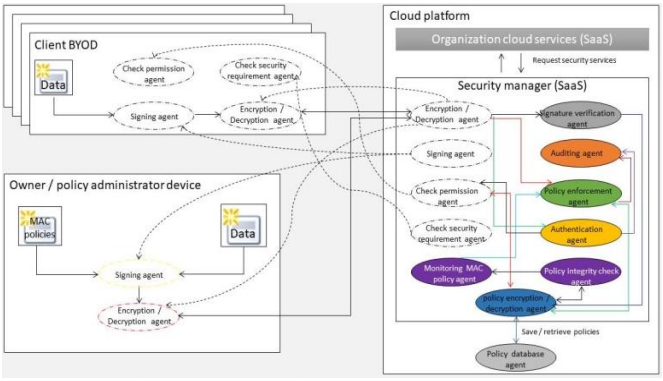

Fig. 2. Our proposed framework for the BYOD and cloud environment

A. Owner / Policy Administrator Device

Whoever is responsible for BYOD user access control policy, controls the owner/policy administrator device. This can be the Chief Security Officer (CSO), policy administrator or an organization's owner. The device can be either a personal device or PC with a trusted operating system like Security-Enhanced Linux (SELinux) so they can set security classification levels and the initial data for user access control.

1) MAC Policy

The Mandatory Access Control (MAC) policy dictates strict access limits that are difficult to bypass, either intentionally or unintentionally. Using a MAC policy is effective as it assigns a clearance level to every user. It does this by establishing what each user can and cannot access within the system using JavaScript Object Notation language (JSON). There are four categories for users (subjects) and resources (objects), and these are top secret, secret, confidential, and unclassified. The policy administrator determines the user and resource security classification levels according to the MAC. The JSON file and data is encrypted and signed after the data is digitally signed. The following is an example of a JSON file:

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

2) DATA:

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

B. Security Manager

The security manager is at the core of the proposed framework. Its function is to manage all the components required for the MAC policy to operate. It is located in the cloud and operates when called on by a SaaS. The framework has four functions: checking BYOD device security, enforcing the access control policy, working with independent platforms, and securing the access control policy. It works in conjunction with the 11 agents.

1) Controller Agent

The controller agent is static and manages all other agents. It contains the Application Programming Interface (API) that allows it to communicate with other SaaS in the cloud. The controller agent creates instances from mobile agents and sends them to devices using individual IP addresses.

2) Check Security Requirement Agent

The controller agent creates the check security requirement agent. Its purpose is to check all connected devices using an organization's SaaS in the cloud. The check security agent checks whether BYODs meet company security policy requirements for being a trusted device. It does this by checking for up-to-date antivirus software, fingerprints, and a VPN connection and installs an agent manager.

This research uses the requirement for an up-to-date antivirus application as an example. When a device does not have updated antivirus installed, the check security agent provides the user a summary. It will summarize what actions the user needs to take for a BYOD to comply with the organization's security requirements.

3) Authentication Agent

Once a device meets security policy requirements, the authentication agent starts. Every user needs a unique identity. The authentication agent validates a user's identity for access

to the system using two types of authentication for extra security.

4) Check Permission Agent

Once the authentication agent finishes its verification, the check permission agent searches the database for the security classification assigned to the username. The agent sends the username to the relevant personal device to make a preliminary decision about granting access. It then implements the MAC policy to authenticate the username against the security classification contained within the MAC policy to make a final decision to grant access or not.

The check permission agent functions to speed up the process if user access is denied before it sends a request to the cloud. It also displays to users their permissions when accessing specific resources. For example, users will see permission details such as read only, read and write, against each file when the system grants access. Once a user has access at this level, the next check occurs in the cloud by the 'policy enforcement agent'.

5) Signing and Signature Verification Agents

Signing and signature verification agents are mobile agents. They check a system access request comes from a known user without being modified during transit. It generates digital signatures for every JSON policy file and data requests from data owners or BYOD users (Fig. 3). The signature verification agent within the security manager verifies the digital signature. It compares the decrypted hash value with the original JSON policy and initiated generated hash to verify it is the same. When the values are equal, the message has not been modified.

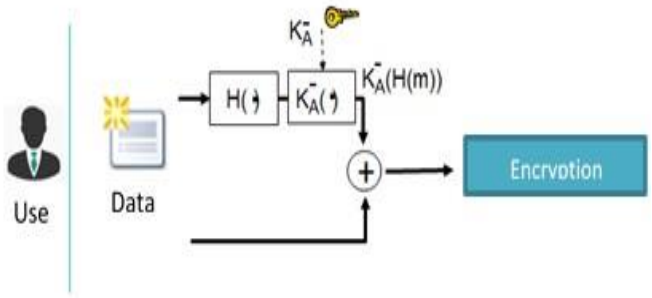

Fig. 3. Generating the digital signature in a BYOD device

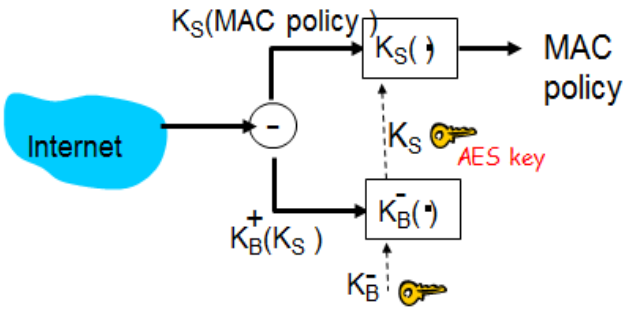

Fig. 4. Decryption of the MAC policy

6) Encryption and Decryption Agent

The encryption and decryption agent makes sure only authorized users and agents access and read the information transmitted. Its function is to keep the information in the message secret. It does this by encrypting and decrypting all transmissions traveling between the security manager and user devices. This agent converts messages into an unreadable format to transmit them. It then reverses the process to convert the messages into a readable format for the user. The agent encrypts messages using an asymmetric algorithm (also known as public-key cryptography). During transmission it exchanges this for a symmetric key (which is the Advanced Encryption Standard (AES)) to decrypt the MAC policy (Fig. 4).

7) Policy Enforcement Agent

The policy enforcement agent is static and its primary function is to enforce access control policies to determine who has access to the cloud. Its purpose is to strengthen access control. The policy enforcement agent implements the MAC policy using the Bell–LaPadula model (Fig. 5) to match the relevant user classification level. It uses the classification level in conjunction with the 'check permission agent' to verify a user has legitimate access and transmissions were not modified during the process.

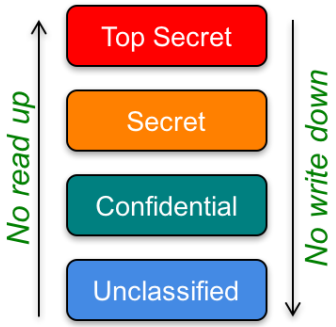

Fig. 5. Bell–LaPadula model

8) Policy Monitoring and Integrity Check Agents

The policy enforcement agent saves a copy of the first hash value generated/updated by the owner. It continuously uses this as a comparison with newly generated hash values for the same MAC policy. These should all be identical. Policy monitoring and integrity checking agents check for modifications to a MAC policy during transmission and has only been sent by the policy administrator. It informs the policy administrator and controller when there is a security breach.

9) Auditing Agent

The auditing agent is static. Its function is to record all successful and failed attempts to access the system. It records all policy enforcement agent decisions about grant and deny access decisions. The auditing agent records username, date, time, access request to what resources, and the decision. This assists the policy administrator to monitor, analyze, and manage regulatory compliance, understand system access denials, and perform disaster recovery to develop the system.

10)Policy Encryption and Decryption Agent

The policy encryption and decryption agent encrypts and decrypts all the data it transmits. It uses the symmetric

encryption algorithm (AES) to protect information during the storage and retrieval phases when communicating with the access control database.

11)Policy Database Agent

The policy database agent is static and communicates with other Databases as a Service (DBAAS), database management systems (DBMSs), or distributed database management systems (DDBMSs). This agent exchanges the data as it transmits across different software architecture styles and patterns.

C. Client Byod Device

When a client uses their BYOD to access the cloud environment, the check security requirement agent verifies it meets security policy requirements. Once the BYOD passes and the security agent grant access, three other agents perform their functions when signing in. These are the encryption and decryption, check permission, and signing agents. Clients are not restricted to working from one location or to 'hours of work'. They can work from anywhere at any time. People can use any device as long as it meets security requirements. Clients create and share data according to their classification levels, which the owner has to approve (or reject). The owner grants access by accepting the request.

Sequence diagrams for the proposed access control framework are broken into seven sub-frameworks based on the main tasks. We explain the most important tasks, which are: creating and modifying policies or data by policy administrators; clients creating and modifying data; and monitoring the MAC policy in the security manager.

1) Creating and Modifying Policies and Data

Fig. 6 shows the process for creating and modifying policies and data. To start, the signing agent adds a digital signature to the message when a policy maker creates a new or modifies existing policies or data through the user interface (Fig.6). Then the encryption and decryption agent encrypts the message to transmit it and decrypts it into a readable format for the receiver. A signature verification agent verifies the digital signature. Then the policy enforcement agent implements the MAC policy to accept or deny illegitimate access requests. There are 'save' mechanisms for both the policy encryption and decryption, and the policy database agents. This is so the user can save new and amended policies and data to the system. The 'ack' method in both agents confirms the save process. Finally, the auditing agent records full details of final decisions made by the policy enforcement agent.

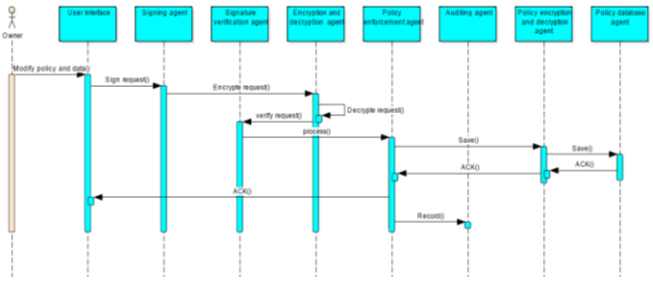

Fig. 6. Sequential diagram for creating and modifying policies or data by policy administrators

When clients try to create or modify data, the permission method makes a preliminary decision about granting access based on the username clearances and security classifications. If permission is granted, the 'signing agent' adds a digital signature to the data (Fig. 7). All data is now encrypted by the encryption and decryption agent before being transmitted across the internet. The agent then decrypts the data when received in the cloud. Next the signature verification agent verifies the digital signature. The policy enforcement agent implements the MAC policy, which denies any illegitimate access requests. The policy encryption and decryption agent and policy database agent save the data. The ack method in each agent confirms the data saving process. Finally, the auditing agent records full details of final decisions made by the policy enforcement agent or by the check permission agent during the preliminary decision.

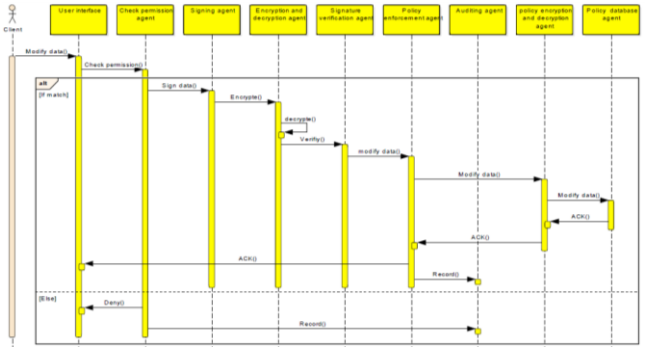

Fig. 7. Sequential diagram for creating and modifying data by clients

2) Monitoring the Mac Policy

Fig. 8 shows the process of monitoring the MAC in the security manager. Monitoring the MAC is the main function for protecting the integrity of policies during the processing and storage phases. This process starts with the controller agent activating the policy integrity check agent. It retrieves the policy from the database using the request hash key method. The reply hash key generates a hash value from the policies requested. The value is sent to the monitoring MAC policy agent for comparison with the original one. If the values match, the process continuously repeats. When they do not match, the monitoring MAC policy agent sends an error message to the controller agent to cease authentication. It records the issue, deletes the existing policies, and sends a message to the owner.

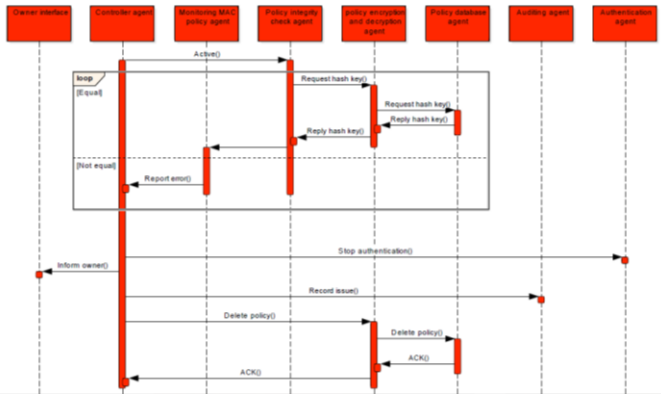

Fig. 8. Sequential diagram for monitoring the MAC policy in the security manager

IV. 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 (Fig. 9).

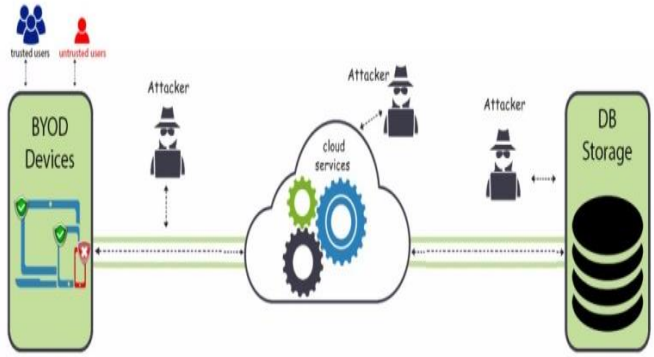

Fig. 9. Potential attacks that may occur in the cloud and BYOD environment

We classified the test into four main sub-tasks. First, trusted and untrusted users and devices testing. Second, access control policy testing. Third, performance and scalability testing. Finally, integrity testing.

1) Trusted and Untrusted Users and Devices Testing

Four different cases can be used to test trusted and untrusted users or devices. These cases cover the possible situations that may occur when users use their BYODs, as seen in (Table 3). These cases are:

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 legitimate resources.

Case 3: The use of a trusted device with trusted users who want to access illegitimate resources.

Case 4: The use of a trusted device with untrusted users.

TABLE III. Different Cases Of Trusted And Untrusted Users Or Devices

Situation of different cases	Trusted devices	Untrusted devices
Trusted users access legitimately	Case 2	Case 1
Trusted users access illegitimately	Case 3	Case 1
Untrusted users	Case 4	Case 1

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

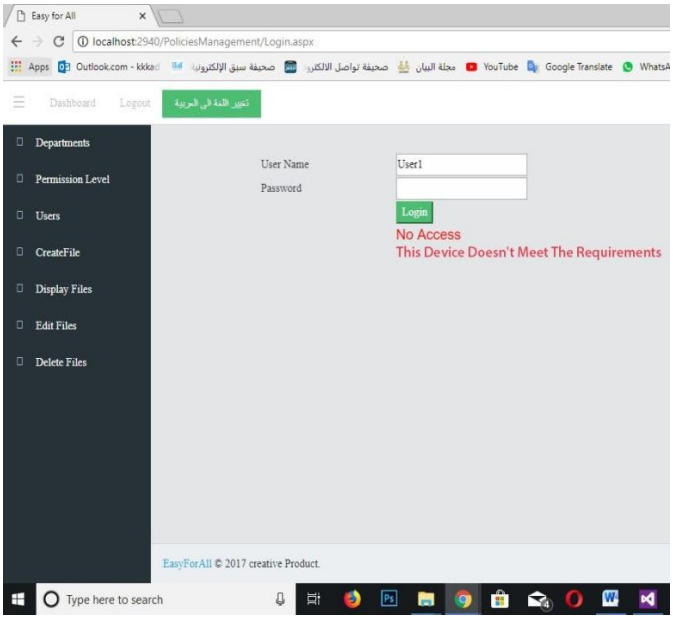

Fig. 10. Interface showing the untrusted BYOD

For the second case, 'check security requirement agent' allowed the device to connect to Google cloud because it is a trusted device. Both 'check permission agent' and 'policy enforcement agent' allowed trusted users to access wanted resources. For case three, the system detects users that want to access illegitimate resources, as shown in (Fig. 11), by verifying the MAC security classification level of the user and comparing it with the security classification level of the wanted resource using the Bell–LaPadula model to gain access. The final case is for untrusted users (i.e., users who do not have permission to access the system and certainly do not have a MAC security classification level). 'Authentication agent' can discover these users and prevent them from accessing the system, as shown in (Fig. 12).

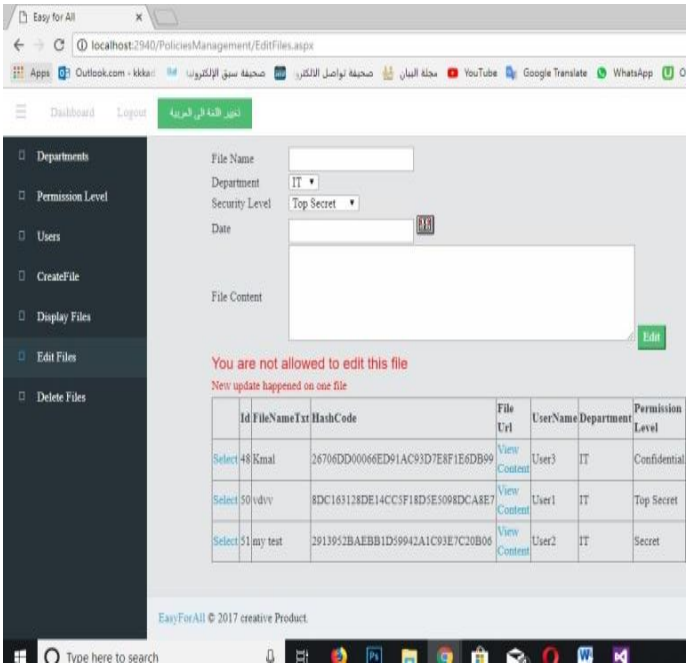

Fig. 11. Interface showing denied access to illegitimate resources

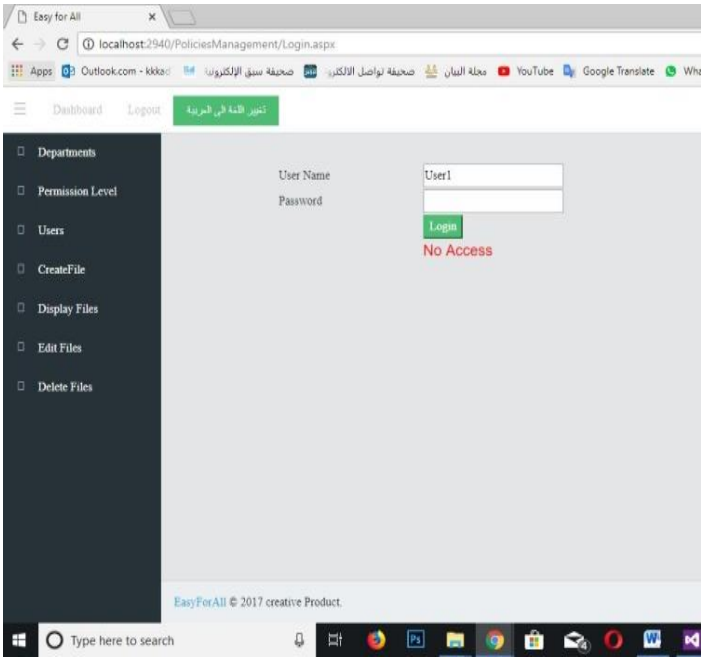

Fig. 12. Interface showing prevention of untrusted user access to the system

2) Access Control Policy Testing

A different access control policy has been tested by simulating different and complex attacks during the transfer, process, and storage phases. During the process and storage phases, the proposed framework faced 10 attacks that modified the access control policy.

Five of them modified during the processing phase, and five of them modified in the database. The hash value changed and was detected by policy monitoring and integrity check agents, as shown in (Fig. 13).

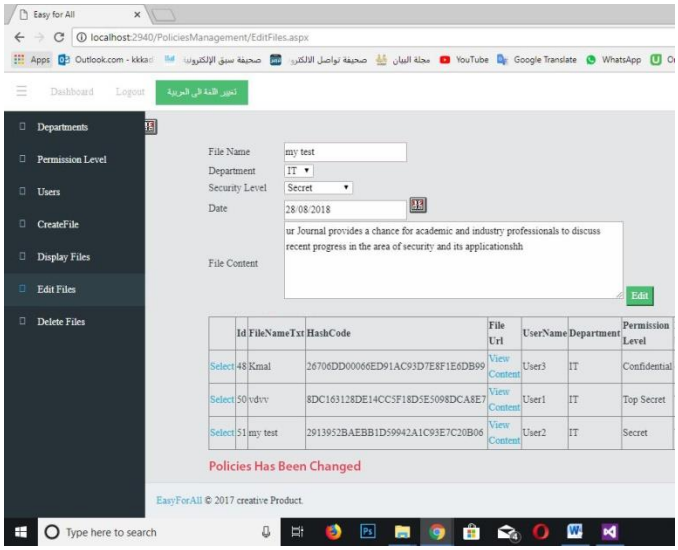

Fig. 13. Interface showing the detection of the changed MAC policy

During the transfer phase, we tested the 20 accesses of the control policy with different characteristics. Five of them had the correct digital signatures, five of them had 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 (Fig. 14).

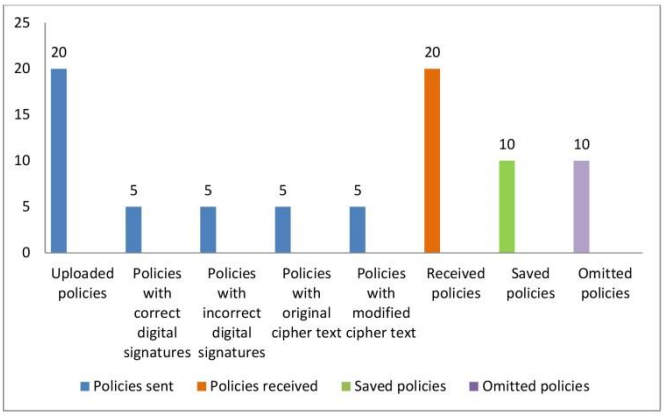

Fig. 14. Statistics shows the number of uploaded policies, received policies, saved policies, and rejected policies

3) Performance and Scalability Testing

To measure performance and scalability, we used different available software based on the required test. Visual Studio 2017 has some useful built-in testing tools that we used to measure the CPU and memory usage. The Google cloud platform also has some useful testing tools for measuring traffic, load, CPU and memory usage, and more. We also used the JMeter tool to test scalability because it is a free open source tool specifically for this type of testing. Below are the results of these different tests with some comments about each test. The discussion and evaluation of the results are in the next chapter. First, we measured the performance with different numbers of users ranging from 1 to 1000 users for the access control enforcement function, as shown in (Fig. 15).

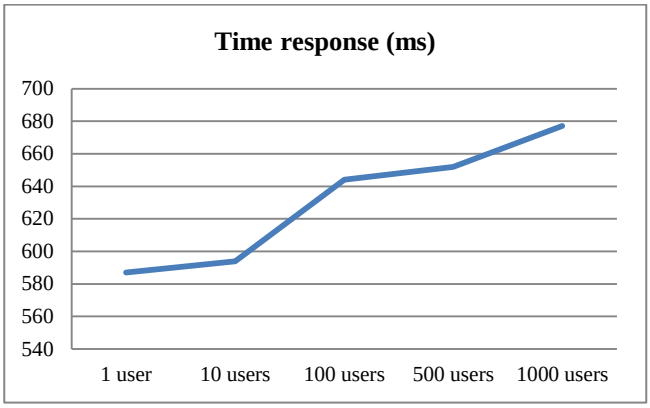

Fig. 15. Performance testing with different numbers of users for the enforcement access function

Fig. 16 shows the time response for each function in the proposed framework, including the time for saving and retrieving data from the database. This test was done in local machines. In addition, we did not calculate the travel time between different machines.

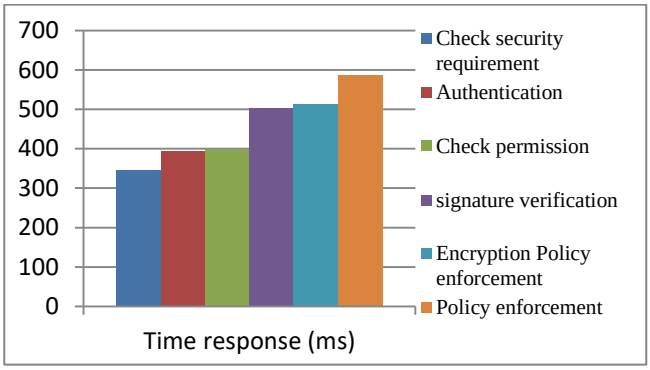

Fig. 16. Time response for each function in the proposed framework

The next test shows the time response for access allowed by the policy enforcement agent and access denied by the check permission agent after the authentication phase and setting up the policies, as shown in (Fig. 17).

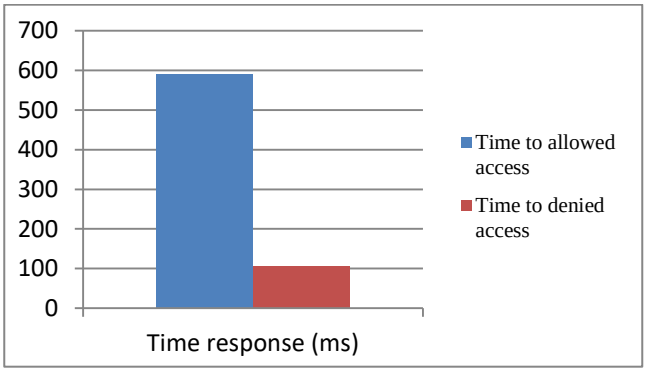

Fig. 17. Time response for allowed and denied access decision

We used a LOADIMPACT tool to test the load time in the cloud when the number of users is increased as seen in (Fig. 18)

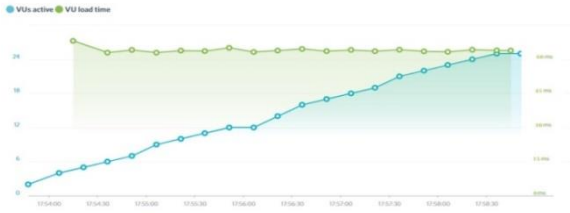

Fig. 18. Load time test for the framework in the cloud with increase of number of users

4) Integrity Testing

Over three weeks, we found 2,860 access requests in the logs. Twenty policy attacks occurred, none of which were successful since they were detected by the system. To measure integrity for a particular type of attack, we need to know the probability that an attack of this type will occur within a given time [29]. The integrity attack is defined as:

Integrity attack = 1 – threat attack (1 – security attack)

The integrity of the software product, integrity, is defined to be the sum, over all attack types, of integrity attack:

Integrity = Σ attack (integrity attack)

In our case, the threat attack is (20/2860) = 0.006993006993007 and the security attack is (0/20) = 0.00, so the integrity is (1- 0.006993006993007 * (1 – 0.00)) = 0.99300699300699 ≈ 1 * 100 = 100%.

V. Discussion And Evaluation

The process of verifying and validating was completed successfully, as planned, by using white and black boxes testing with no faults, errors, or failures in the system. The results can be explained as follows. First, the proposed framework was able to differentiate between trusted and untrusted devices and between trusted and untrusted users. It prevented untrusted devices from connecting to the cloud and prevented untrusted users from accessing the system. It also enforced the access control policies and provided access to legitimate users only. Second, the proposed framework was able to detect attacks that faced access control policies during the transfer, process, and storage phases. It rejected the policies that had been modified after informing the owner of the system.

Third, the performance tests showed a slight increase in the time response when the number of people increased during the process of enforcing access control policies in the local machine. This kind of test examines the scalability of the system. The result is normal due to specific resource consumption, such as CPU and memories. However, the same test was done in the cloud by a LOADIMPACT tool and showed no increase in the time response. This is because the resources in the cloud are scalable, which means the cloud is able to increase the workload on its current hardware resource on demand with an increase in the amount of billing. Fourth, the JMeter testing tool showed the time response for each function in the system. The policy enforcement had the highest time response due to the comparison between the security classification levels of the subjects and objects after retrieval of

these data from the database. The encryption and decryption functions had high time responses because of the amount of time the asymmetric algorithm needed to encrypt and decrypt the messages.

Fourth, we reduced the time needed to make the final decision when an illegitimate request occurs due to the functionality of the check permission agent in the same BYOD. However, in the case of legitimate access, it takes more time because the decision comes from the policy enforcement agent in the cloud. Fifth, these tests were performed using Intel Core (TM) i7 -5500U CPU (2.40 GHz) and 8.0 GB DDR3 memory, which shows low performance for one user. Finally, the integrity of the system is high due to its detection of the attacks, which were unsuccessful.

VI. 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 increase the performance of allowing access decision and enhance the current framework to support federated cloud computing.

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. 78,2016, pp. 179-184.

[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, 2015, pp. 38-54.

[6] P. Beckett, 'BYOD–popular and problematic,' Network Security, vol. 2014, pp. 7-9.

[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, 2015, pp. 281-297.

[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, 2017, pp. 53-62.

[9] S. Blizzard, 'Coming full circle: are there benefits to BYOD?,' Computer Fraud & Security, vol. 2015, 2015, pp. 18-20.

[10] B. Morrow, 'BYOD security challenges: control and protect your most sensitive data,' Network Security, vol. 2012, pp. 5-8.

[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. (2014). Security Report. Available: https://www.checkpoint.com/documents/ebooks/security-report-2014/files/assets/common/downloads/Check%20Point%20Security%20Report%202014.pdf. [Accessed: 10- Oct- 2018].

[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, 2017, pp. 69-75.

[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, 2015, pp. 81-99.

[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, 2013, pp. 5-6.

[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, 2015, pp. 83-95.

[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, 2017, pp. 44-54.

[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, 2015, pp. 129-136.

[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, 2015, pp. 511-516.

[29] G. G. Schulmeyer and J. I. McManus, Handbook of software quality assurance: Van Nostrand Reinhold Co., 1992.