Parkinson and Khana *Cybersecurity (2022)5:6*
https://doi.org/10.1186/s42400-022-00112-1

Cybersecurity

RESEARCH Open Access

Identifying high-risk over-entitlement in access control policies using fuzzy logic

Simon Parkinson* and Saad Khana

Abstract

Analysing access control policies is an essential process for ensuring over-prescribed permissions are identified and removed. This is a time-consuming and knowledge-intensive process, largely because there is a wealth of policy information that needs to be manually examined. Furthermore, there is no standard definition of what constitutes an over-entitled permission within an organisation's access control policy, making it not possible to develop automated rule-based approaches. It is often the case that over-entitled permissions are subjective to an organisation's role-based structure, where access is be divided and managed based on different employee needs. In this context, an irregular permission could be one where an employee has frequently changed roles, thus accumulating a wide-ranging set of permissions. There is no *one size fits all* approach to identifying permissions where an employee is receiving more permission than is necessary, and it is necessary to examine them in the context of the organisation to establish their individual *risk*. Risk is not a binary measure and, in this work, an approach is built using Fuzzy Logic to determine an overall risk rating, which can then be used to make a more informed decision as to whether a user is over-entitled and presenting risk to the organisation. This requires the exploratory use of establishing resource *sensitivity* and user *trust* as measures to determine a risk rating. The paper presents a generic solution, which has been implemented to perform experimental analysis on Microsoft's New Technology File System to show how this works in practice. A simulation using expert knowledge for comparison is then performed to demonstrate how effective it is at helping the user identify potential irregular permissions.

Keywords: Fuzzy control, Fuzzy systems, Security, Access control policies, Security analysis, Risk, Fuzzy logic, Risk-adaptive access control

Introduction

Access control systems are an integral mechanism within computing systems, whereby access to resources are regulated to ensure those deemed to be sensitive are only accessed by authorised users (Sandhu and Samarati 1994). Access control systems often provide many levels of access, going beyond simply granting or denying access. They provide many different granularities (e.g., read, write, etc.) of access to accommodate the many different potential security situations that may arise within an organisation (Ouaddah et al. 2017). There are many different types of access control models that are widely used (Samarati and de Vimercati 2000). For example, role-based access control centres around providing permission levels depending on a user's business function within the organisation (Ferraiolo et al. 2003), whereas Discretionary Access Control systems provide a fine-grained level of control on a per resource basis which can be administered by the user (Osborn et al. 2000; Pfleeger and Pfleeger 2002). The emphasis in the research presented in this paper is on implemented access control systems. A common aspect is that they all provide mechanisms for the user to receive permission from multiple policies, thus resulting in an accumulated enforced *effective* permission for the user. The effective permission describes the level of access granted to a resource for a

*Correspondence: s.parkinson@hud.ac.uk
Department of Computer Science, University of Huddersfield, Huddersfield HD1 3DH, UK

specific user, taking into considering multiple permissions being allocated as well as conflict resolution.

Access control implementations are often audited to review the security policy, with a particular focus on identifying instances of *over-entitlement*, which is where the user has more permission than they need to undertake their role. Over-entitlement can be particularly dangerous. For example, should the user be involved in a security incident or themselves become an adversary, there is the potential for the user to access more resources and therefore cause more damage. This is also true of ransomware when it executes under the user's account and acquire their level of access (Parkinson 2017). The requirement to perform an audit is common amongst all access control implementations, but a significant challenge is that it is not a binary task, as even though permissions that are irregular and potentially anomalous can be identified, they do not necessarily indicate a high-risk situation requiring immediate intervention. Irregular and anomalous permissions can be thought of as those that are noticeably different from the full access control policy (Hu et al. 2013). This could be where permission is relatively different (in occurrence and power) from all others and warrant further investigation. For example, a trustworthy user having incorrect access to a insensitive resource is also of low significance. Determining if the level of access is of potential security concern is by no means a trivial or binary task; It is subjective to the sensitivity of the restricted object, and the trustworthiness of the user, which can combine to give a level of risk.

The use of user trust and resource sensitivity is by no means new to access control systems, with research studies (Ryutov et al. 2005; Ahmed and Alnajem 2012; Atlam et al. 2017) and patents (Salem et al. 2013; Cheng et al. 2016) discussing how they can be utilised. However, determining the trustworthiness of a user and sensitivity of resources is challenging and does require understanding of business activity, user behaviour, and the importance of the secured resource. Furthermore, expertise to determine these factors may also be in short supply, and in some instances not available at all. For example, consider a large organisation with many employees and electronic resources. It is highly unlikely that a single individual possesses sufficient knowledge to understand user trust and resource sensitivity across the entire organisation. However, if someone with such knowledge is available, it is reasonable to suggest that they will still benefit from a technological aid, assisting to improve reliability [i.e., reduce human error (Cheng et al. 2007)] and reduce the required time and effort. As there are different aspects to consider when analysing access control policies, such as user trust, resource sensitivity and permission risk, it is necessary to consider their interaction. It has previously been highlighted that the relationship between these inputs is best suited to being modelled and represented in fuzzy logic, where values such as user trust that are not binary can be represented by a probability of truth (Cheng et al. 2007; Atlam et al. 2019; Bernabe et al. 2016). However, these approaches are on the topic of introducing fuzzy sets into the access control model and not as part of the analysis process. Furthermore, a challenge exists in constructing resource sensitivity and user trust levels and considering their relationship with the access control policy to establish a measure of risk.

In this work, we aim to solve this problem and utilise fuzzy logic in the analysis process through the development of a novel technique, whereby a user's effective permission on all access-controlled objects is modelled in a risk-based fuzzy model, which is subsequently used for analysis purposes and for detecting implemented permissions which have the highest level of risk. This information, once presented to the user, can be used to determine any security risk, and assist the user in identifying permissions that warrant further consideration. In this work, it is assumed that no knowledge is available within the organisation as to how sensitive resources are and how a user's trust may change depending on their job role. For this reason, an approach is developed to estimate the resource sensitive and user trust based on available system activity information. This paper investigates the research hypothesis of analysing access controls using fuzzy logic, based on automatically extracted measures of user trust and resource sensitivity, enables the easier detection of high-risk and potentially anomalous permission entries.

In this paper, the following novel contributions are presented:

- A model is provided based on fuzzy logic for analysing implemented access control policies, using trust and sensitivity measures to determine a measure of risk for each user-permission-object relationship.
- A technique to estimate trust and sensitivity values, based on available system information alone, which in the case of this research is that extracted from security event information sources.
- Empirical analysis using a systematic methodology to establish the technique's capabilities on simulated directory structures.

The paper is structured as follows: the following section presents a discussion of previous and related work, as well as motivating the underlying research presented in this paper. The section after details the development of

the fuzzy logic-based access control analysis system. This then leads to the implementation of the fuzzy logic system for analysing Microsoft New Technology File System (NTFS). Empirical analysis is then performed whereby testing is performed on simulated file systems. Finally, a conclusion is provided, laying out future research directions.

Related work

Fuzzy logic has previously been utilised in access control systems, where they aim to solve the challenge that implementing access control is often a problematic and error-prone activity. For example, assigning a user to inherit permissions through group membership might result in them acquiring too much or too little permission elsewhere. In previous work, a fuzzy logic-based access control system is presented, whereby users are granted access to objects based on their associated risk and sensitivity ratings (Ni et al. 2010). In principle, the system works by attributing a security score to both user and object, which is then used alongside a fuzzy system to determine an overall risk rating to determine access. Similar work has also been presented to operate in Internet of Things (IoT) architectures (Mahalle et al. 2013). The use of fuzzy logic in access control systems is diverse. Recent implementations include crowd-sourcing environments (Folorunso and Mustapha 2015), wireless body networks (Nekooei et al. 2017), cloud computing (Younis et al. 2014), and IoT health systems (Abie and Balasingham 2012). A key and common aspect of these approaches is that they require prior information used to determine a risk rating. This information is required to be available for the access control systems to function; however, they do surpass more traditional access control systems in terms of configuration and maintenance. Furthermore, the requirement for the administrator to set trust and sensitivity levels could be seen as a burden, but more significantly introduces the potential for the permissions to be set incorrectly in the first instance.

In terms of analysing access control systems, previous works in discovering irregularities in access control systems implemented a binary classification system, whereby access control rules are identified as either normal or irregular. This often involves identifying individual user-permission-object relationships, $UPO \in USER \times PERMS \times OBJECT$. In terms of identifying irregular permissions, the permissions are divided into regular and irregular sets, R and I, respectively. For a *crisp* representation an individual relationship $(u, p, o) \in UPO$ is either a member of the set I or not. This binary representation is modelled in the following indicator function:

$$\chi I(P) = \begin{cases} 1, & (u,p,o) \in I \\ 0, & (u,p,o) \ni I \end{cases} \quad (1)$$

In previous work whereby the inclusion of user-permission-object relationships, (u, p, o), in set I is determined by a measure of irregularity, meaning that an unambiguous lower, l, and upper bound, u, threshold is used to determine the set relationship (Parkinson et al. 2019). This would result in set I containing all those elements where $l \leq I \leq u$.

There is a significant challenge with the binary and unambiguous classification function. This is that in performing a crisp set classification, techniques are derived to determine whether permissions are normal or over-entitled. Due to this binary classification, any uncertainty in determining would result in the incorrect identification of normal permissions, or more significantly, the incorrect identification of over-entitled permissions as normal. Earlier work in identifying file irregularities permissions using statistical technique (χ^2) demonstrates a good level of accuracy (91%) (Parkinson and Crampton 2016). The technique utilises χ^2 statistics and Jenks natural break to determine set boundaries. Other research involved the use of Association Rule Mining to discover rare and potentially over-entitled item sets, which also resulted in a comparable level of accuracy as with the statistical approach (Parkinson et al. 2016). Although these techniques demonstrate a good level of performance, challenge exist when trying to make further improvements. The techniques are over-sensitive, meaning that they identify many false positives (normal permissions incorrectly identified). There are also cases where over-entitled permissions are not correctly identified in instances such as where the over-entitled permissions are not statistically different from normal. Further research has presented the use of these techniques to identify a specific type of irregularity (permission creep) with an average accuracy greater than 90% (Parkinson et al. 2019). This level of accuracy is still significant considering that the techniques have no prior knowledge of what constituents an irregularity; however, there is strong motivation to identify all instances when considering the necessity to maintain resource security.

The consideration of access control and risk has long since been established. For example, in early research fuzzy logic was used to determine the risk of users within systems (Friedlob and Schleifer 1999). However, there is little works in terms of using fuzzy logic for analysis. An expert system has been developed and tested for analysing system security using fuzzy logic, with an example in determining user risk based on password strength (Kozhakhmet et al. 2012). Other research

has demonstrated the potential of using a Fuzzy based approach for analysing buffer overflow vulnerabilities (Shahriar and Zulkernine 2011). Based on the strength of research into using fuzzy logic in access control systems, and the absence of research in analysing traditional access control mechanisms using a fuzzy approach, this paper investigates the potential of analysing access control systems using fuzzy logic.

Modelling

There are many different types of access control models utilised in computing systems, which offer different properties in regard to policy administration and enforcement. Mandatory Access Control (MAC) is suited to safety critical systems whereby a central authority administers and enforces the policy. The properties of MAC make it suitable for being used in military systems (Ray and Kumar 2006). Role-Based Access Control (RBAC) systems enables the restriction of access based on user's role within the host organisation, such as management m finance, etc. Sandhu et al. (1996). Discretionary Access Control (DAC) required administration and the enforcement of access control on a per user basis. A central feature to DAC is that resource owners can assign permissions to other users at their discretion. Researchers have studied the safety and complexity of DAC systems and developed algorithms for determining safety (Li and Tripunitara 2005).

A commonality across all systems is that the user will receive an *effective* permission, which is essentially the resolved permission on a given resource, accounting for role inheritance, conflicts, etc. This work is motivated by end-user challenges in analysing access control systems; thus, we are interested in obtaining the effective permission, irrespective of what access control model is used. The reader should note that this research was motivated by challenges facing analysing Microsoft's NTFS file system permissions, which is a DAC system combined with MAC, and the combination of the enables the creation of flexible policies, which can be used to represent an RBAC system.

In this section, the effective permission model is presented. The objects (also known as resources), $OBJECTS = \{o_1, o_2, \ldots o_n\}$, represent components within the system that require controlled access. An object could, for example, be a file system resource, a printer, software service, etc. The users, $USERS = \{u_1, u_2, \ldots, u_n\}$ represents those interacting with the system and are granted perspective permissions. For example, a user could be a user, a process, etc. The level of permission, P, will often be described by a series of permissions attributes, $PERMS = \{p_1, p_2, \ldots, p_n\}$. The individual permissions will differ dependent on the underlying access control system. In this paper, we are not concerned with how the permissions are allocated to the user, and we assume that in the access control system there is a mechanism to determine a user's effective permission on an object. The effective permission is the relationships set of:

- $UPO \subset USERS \times PERMS \times OBJECTS$

which is the set of user-role assignments. While conducting an audit, all effective permissions are calculated by considering all user and object permutations; however, entries are not created where there is no implemented permission on an object for a given user.

Each user within the system is assigned a *trust* value, which is a numeric score between 0 and 1. Similarly, each object is assigned a *sensitivity* value. Each permission is also assigned a *power* rating, which is also a score between 0 and 1 as to the capability of the permission. The combination of these values is then used to calculate a final *risk* rating.

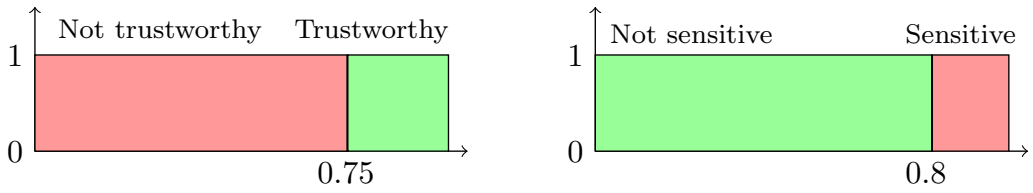
- $UPO \subset USERS \times PERMS \times OBJECTS \rightarrow [0, 1]$

The user-permission-object (UPO) mapping of set items in the form of $(u, p, o), \mu(u, p, o)$ where $u \in USERS$, $p \in PERMS$, and $o \in OBJECTS$. The function $\mu(u, p, o) \rightarrow [0, 1]$ is the expressed as real unit interval between 0 and 1. The function represents the final risk rating, calculated using $\mu(u)$, $\mu(p)$, $\mu(o)$ which represent the real unit interval scores between 0 and 1 for user trust, permission power, and object sensitivity, respectively.

When considering these values in establishing an overall risk rating ($\mu(u, p, o)$), it is important to consider the threshold at which a regular permission is identified as irregular and warrants further investigation. As an illustrative aid, Fig. 1a details the binary relationship of whether a user is trustworthy or not based on the threshold of 0.75. Figure 1b illustrates where a resource is deemed to be sensitive at the threshold of 0.8. Although these figures are arbitrary for the example, it does demonstrate that these two measures (trust and sensitivity) are based on binary classification. Using these two values together can help determine whether an effective permission is determined as putting the underlying object as risk.

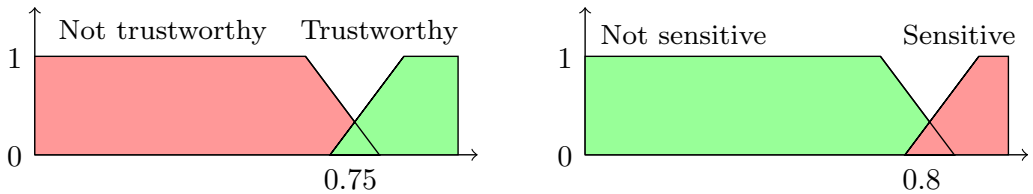
Fuzzification

Although evaluating an effective permission's risk rating against a threshold is somewhat useful, it does not adequately describe the relationship between a user's trustworthiness and an object's sensitivity. More specifically,



(a) Trust classification **(b)** Sensitivity classification

Fig. 1 Binary view of considering user trustworthiness and object sensitivity



(a) Trust classification **(b)** Sensitivity classification

Fig. 2 Fuzzy view of considering user trustworthiness and object sensitivity

the binary representation determining that the metrics are either true or false (0 or 1) lacks sufficient expressiveness. For example, a user does not go from being trustworthy to untrustworthy precisely at the threshold, largely due to the uncertainty with estimation. In practice, a fuzzy representation more accurately models the relationships. Figure 2a illustrates a fuzzy relationship between user trust and Fig. 2b illustrates object sensitivity. It is evident that although the same arbitrary thresholds are used as in the binary classification example (0.75 for trust and 0.8 for sensitivity), the classification is no longer binary and the partial truth, i.e., a user has lower trust but is not completely untrustworthy, is adequately modelled. In this work, we utilise a trapezoidal function to model the different sets. In terms of the construction of the trapezoidal function coverage, 0 to 0.2 would occupy the left ascending section, 0.2 to 0.8 the flat middle section, and 0.8 to 1.0 for the descending right section.

The example in Fig. 2 does represent the cross-over relationship between a user being trustworthy and data as being sensitive. It also only represents two potential classes for both trust and sensitivity; however, it is widely acknowledged that there are many levels of trust and sensitivity (Mhetre et al. 2016). In this paper, we adopt an incremental hierarchy of trust, meaning that a user must meet the criteria of one level before they can progress. We adopt the classification as demonstrated in Neil Normal Group report on Hierarchy of Trust: The 5 Experiential Levels of Commitment (Sherwin 2016), which is built upon Abraham Maslow's hierarchy of needs (McLeod 2007). The specific five levels of trust are widely adopted as the *pyramid of trust*, as well as introducing a sixth level to represent that no trust has yet been established. The pyramid is typically used from a user's perspective, but the viewpoint is changed in this work to be that of the employer, system or data owner. The trust levels are:

1. **No trust** as trust has yet to be established for the specific user. For example, this could be a new user who has not yet gone through basic IT training to ensure they understand the organisation's expectations of computer use.
2. **Baseline** relevance and trust that needs can be met. For example, this could mean that a user has some knowledge of data sensitivity concerning the organisation's activities and has undertaken basic training.
3. **Preferential** trust over other options is where a user can determine the most applicable actions to take in regards to correct system use. I.e., they can make conscious decisions to ensure correct system and resource use.
4. Can be trusted with **Personal** information is where the users have demonstrated with a proven track record that they are trustworthy with personal data within the system.
5. Can be trusted with **Sensitive** information is where users can be trusted with resources of a sensitive nature, for example, business-critical documentation.

6. **Committed** to an ongoing trust relationship means that users have demonstrated that they will always act responsibly and in the best interest of the organisation, respecting resource trust.

In terms of sensitivity, five levels of sensitivity are used in this paper. Although it is possible to establish many levels of sensitivity, a five-tier hierarchy is adopted to avoid introducing unnecessary complications by having too many levels. Researchers have proposed the use of resource sensitivity levels to be used in access control systems, authentication and authorisation services (Gaddam et al. 2014). However, although all the published material describes and presents the use of sensitivity levels, they do not explicitly define what levels are to be used. In one recent article, the authors describe that resource sensitivity is identified based upon usage patterns of resources, without any prior knowledge of the resource's content (Park et al. 2016). Unfortunately, the fine detail of their approach is not available. Other research and guidance utilise the phrase of 'Data Classification' in terms of placing it into different classes of sensitivity, based on factors such as their usage (Shaikh and Sasikumar 2015; Lu et al. 2015). In previous research, authors have mined sensitivity levels in large commercial infrastructure, arriving at 11 clearly separate sensitivity levels based on text analysis and document content (Park et al. 2011). A common aspect of these works is that they classify resources sensitivity into discrete levels. For this research, the following five levels of resource sensitivity are adopted:

1. **Unclassified** is that data has no sensitivity classification, which could either be through no prior consideration or a deliberate assignment that the resource does not need to be classified. This could, for example, be data that is already in the public domain, such as marketing information.
2. **Normal** represents resources that are not sensitive, yet there should remain a basic level of access control to minimise taking unnecessary risk.
3. **Moderately** sensitive resources are those that should have their access controlled, but is not business-critical nor requires rigorous enforcement.
4. **Highly** sensitive resources need strict access control which should be rigorously enforced and monitored. An example of such resources could be an organisation's employee personal data which must not be released outside of the organisation.
5. **Ultra**-sensitive where the sensitivity of a resource is such that is cannot and must not be viewed by any user without the necessary permissions. This could, for example, be that legal data is access protected to ensure a legal case is not put into jeopardy

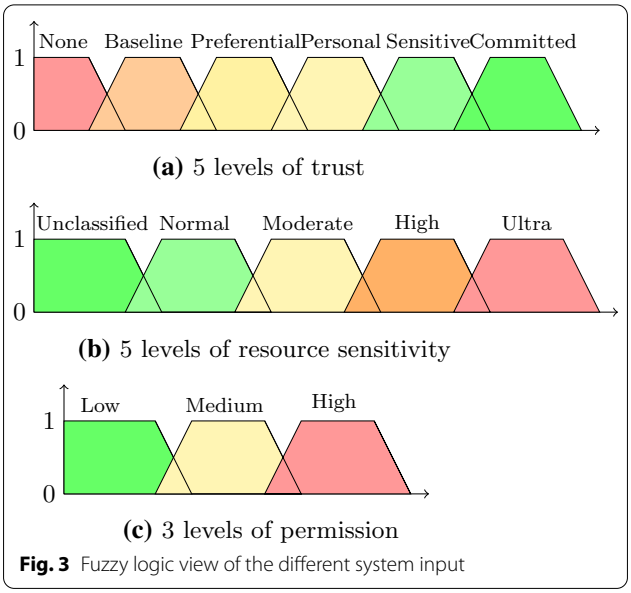


Fig. 3 Fuzzy logic view of the different system input

Both Fig. 3a, b provide a graphical illustration of how the different trust and sensitivity levels are modelled. The final contributor to the overall risk value is the permission itself. In this paper, an approach is adopted whereby individual permission attributes have an associated *power* rating. This power rating is used to assess the potential impact of a user's permission. For example, a delete permission would be high, whereas the ability to read permission attributes is low. Other researchers have utilised a permission rating alongside user trust and resource sensitivity. For example, in one piece of work, a game-theory approach is taken using user trust and permission risk (Helil et al. 2017).

In this work, a three-stage hierarchy is adopted with low, medium and high-power ratings, which are represented by low, medium, and high, respectively. Figure 3c provides a graphical illustration of the three levels and how they are represented in a fuzzy system. The adopted approach is similar to that presented in other research, where permission risk has been successfully modelled into three discrete levels (Rahmati et al. 2018). In terms of accumulation of attribute power ratings, the power rating is calculated on the effective permission and the most expressive power rating is used. For example, a power rating of high is used if the user can read, write, and delete. The following list explains the three levels of power rating used in this research:

1. **Low** represents permissions that are of little security concern with regards to the underlying data. This could, for example, be the ability to read a resource's permission attributes.

2. **Medium** represents permissions that are a security concern but are not likely to cause a security-related incident should the user remain trustworthy. An example could be the ability to read the resource's contents.
3. **High** power represents permissions that have great potential to impact on the resource in terms of confidentiality, integrity, and availability (CIA). An example is the ability to change a resource's security permissions.

The graphical representations presented in Fig. 3a, b demonstrate the overlap between the different levels of trust and sensitivity. When the continuous values of trust and sensitivity are processed by the system, it is necessary to determine which set they reside within and thus fuzzify the continuous value into a linguistic representation. Fuzzification is the process of converting these numeric input variables to linguistic representations. In the proposed system, fuzzification is performed as the following subsections.

Fuzzy inference process

The first stage is to use the input variables (trust, sensitivity, and power rating) to determine an overall *risk* rating. Risk is defined as a continuous numeric value in the same way as the trust, sensitivity, and power values. However, to specify the membership functions and model the fuzzy system using linguistic terms, it is necessary to define the levels of risk. In this work, three levels of risk have been defined, and these are:

- **Low** risk is where trustworthy users are interacting with resources of a low sensitivity and have a low permission power. Therefore, their permission poses little risk.
- **Medium** risk is where a user with anything other than the highest level of trust is interacting with resources with a moderate sensitivity. Such instances of medium risk may warrant further investigation depending on the organisation's tolerance to risk.
- **High** risk is where users with low levels of trust can access resources with high sensitivity levels. Permissions that are of high-risk are those that require further analysis.

The implemented system utilises a risk matrix detailing the relationship between trust, sensitivity, and permission power inputs and the risk output value. Figure 4 provides a graphical illustration of the final risk rating and the contributing resource sensitivity, permission risk and user trust. Risk is shown by the shade of colour (green low-risk to red for high-risk). In this risk model, there

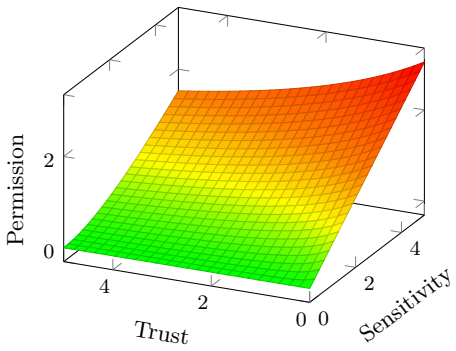


Fig. 4 Membership functions and their relationship to the final risk score

are 6 levels of trust, 5 levels of sensitivity, 3 levels of permission power. In total, this would result in 90 linguistic *if-then* rules. For example, using the table in conjunction with the previously described levels of sensitivity and trust would result in the following three example rules:

IF (Notrust & Unclassified & High) THEN risk = high
IF (Baseline & Unclassified & Medium) THEN risk = medium
IF (Preferential & Unclassified & Low) THEN risk = low

Defuzzification

The final stage of the fuzzy process is to convert the linguistic rules back to a single output, which is the measure of *risk*. In the presented technique, the Mean of Maxima method is implemented (Patyra and Mlynek 2012). In this method, the defuzzified value is taken as the element with the highest membership values. When there are more than one element having maximum membership values, the mean value of the maxima is taken.

The process adopted is as follows: Let I be a fuzzy set with membership function $\varphi(x)$ defined over $x \in X$, where X is a universe of discourse. The defuzzified value, x^*, of a fuzzy set and is defined as:

$$x^* \frac{\sum x_i \in M x_i}{|M|} \tag{2}$$

Here, $M = \{x_i | \varphi(x_i)\}$ is equal to the height of the fuzzy set I and $|M|$ is the cardinality of the set M.

The following presents an example output from using this technique for file system access controls:

Administrator, 0.83, Research\Homes2, 0.80, 1, 0.5

These values constitute the following comma-separated values in order: the username, user trust value, object name, object sensitivity value, permission risk value, and final risk classification.

Note that in the aforementioned example, user trust, object sensitivity, and permission risk have been converted into percentages. For example, user trust of 0.83 is 5 (sensitive), object sensitivity of 0.8 is 4 (highly), and a permission risk score of 1 is 3 (high). The output score is 0.5, which is 2 (medium risk) as the user has a high level of trust on resources of high sensitivity.

Establishing trust and sensitivity values

In the previous modelling section, access control systems are modelled in terms of three distinct numeric values: (1) resource sensitivity, (2) user trust, and (3) permission power. These three parts are then used as input to the system to determine an overall 'risk' rating. Although possible to ask the user to input these values, it would be an exhaustive process on systems with large quantities of resources and users. Therefore, in this section, we present a generic mechanism to establish these three values from available system information. Other researchers briefly present the idea of identifying resource sensitivity from resource use patters (Park et al. 2016). In this research, we adopt a similar process for both trust and sensitivity using available information provided through event logging mechanisms. In developing an autonomous mechanism, it is necessary to make the following assumptions which are described and justified in the following sections. As this work has been performed for Microsoft NTFS policies, we utilise Windows Event Logging mechanisms.

The section focuses on extracting information from available system sources, which are those shared by the majority of Security Information and Event Management (SIEM) systems (Parkinson and Khan 2018; Parkinson et al. 2018; Khan and Parkinson 2018, 2019). In SIEM systems, an event log, E, consists of a series of events ($E = \{e_1, e_2, \ldots, e_n\}$) where each individual event is a tuple, $e = \{T, I, O\}$, consisting of a timestamp (T), an event ID (I), and a set of objects denoting the event description $O = \{o_1, o_2, \ldots, o_n\}$. The objects often refer to components of the system (resources, users, etc.). Events can loosely be coupled into three categories: information, warning, and error. Information would often provide routine contextual information that could be of benefit, for example, a user successfully authenticating. A warning could be that something happened that needs to be examined, for example, a user trying to authenticate with an incorrect password. An error is something of a more serious nature, for example, the crashing of a security service. An example in the Microsoft system is that if someone made ten failed login attempts into a server, the security event logs will contain ten events logs with $I = 4625$. Each entry will have the information about the account name, failure reason, date/time, source network address, port, etc. Together with this data, an expert can determine if there was a security breach incident along with its kind and what security measures should be taken to avoid this in future.

In terms of prior knowledge, it is a requirement that the administrator has classified event types to identify those that impact on user trust and resource sensitivity. In the example presented in this paper ("Section Implementation (NTFS)"), events of interest are defined as those of a known security type.

In using events to establish user true and resource sensitivity, the following three assumptions are made based on common security practice:

Assumption 1 A higher resource usage is likely to mean that the resource is more valuable to the organisation and therefore has a higher degree of sensitivity.

To measure sensitivity, we adopt the approach of counting the occurrence of the resource in the event log. This is performed by iterating over the set of events, E, and increasing a counter. Following this, the score is normalised to enable systematic comparison. Although this is a big assumption, it is reasonable to suggest that a resource that is being heavily used is serving an important business function. However, it is also possible for routinely used documents to be of low value (e.g., temporary system files), but it is viewed to be better to overstate a document's sensitivity than under. Previous work has demonstrated the use of sensitivity levels in respect to resource security (Kiedrowicz et al. 2015; Stanik 2017). Furthermore, in previous research, sensitivity levels are established based on the textual contents of documents (Park et al. 2011). However, this technique is not suitable for any other file type. More recent work by the authors did consider usage patterns (Park et al. 2013).

Assumption 2 A user can be regarded as having a lower trust if they have previously attempted or successfully violated a security policy.

This is measured through the number of times a user is reported as being involved with security actions that record log entries of an adverse type. For example, warnings and error messages indicating that a user has unsuccessfully authenticated. From a security perspective, this assumption is strong as any user that has previously been identified as violating a security policy is going to present an elevated threat level to the system. There are many

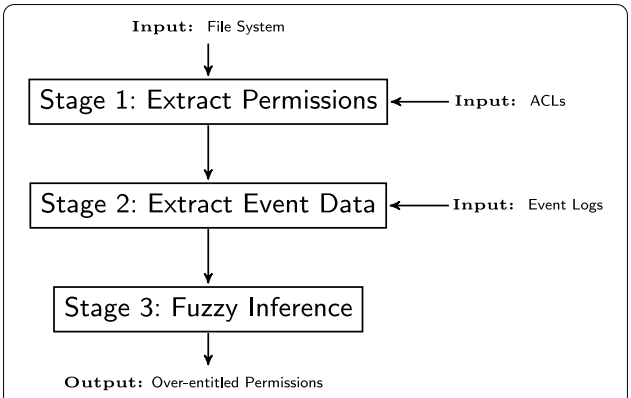


Fig. 5 Schematic illustrating the multiple components and information flow

related research works where user trust is based on a very similar assumption, whereby user trust is established in social media environments (Bodnar et al. 2014), vehicle networks (Zhou et al. 2015), and Ubiquitous computing environments (Leichtenstern et al. 2010).

Assumption 3 A more expressive permission is one that has potential to allow the user to cause greater damage to resources.

Prior knowledge is required in the form of an ordered permission list. The ordered list is then used to assign a numeric score to each attribute, where the most expressive combination adds to the value of 1. For example, in the Microsoft NTFS access control environment, where fourteen individual permissions are available, each permission attribute would be scored as 0.071 and the maximum combination (Full Control) would be scored at 1.0.

Implementation (NTFS)

The fuzzy logic model presented in this paper is implemented and tested for analysing access control models. More specifically, file system access controls used within Microsoft's NTFS. This file system is chosen as the primary target due to its large commercial application and ease of use when creating a realistic test environment. The flexibility of the presented analysis technique ensures that it can easily be utilised for any access control system.

In this research, a software prototype tool is implemented for testing purposes. Figure 5 provides a graphical illustration of the implementation in terms of the different process stages and inputs and outputs. In addition, to help the reader understand the high-level details of the implementation, the following overview description is provided:

- **Stage 1: Extract Permission.** In this stage, the directory structure is processed to establish the permission set of each user resource. The Access Control List (ACL) is examined for each directory and is used to resolve the effective permission for each system user. This involves accumulating multiple policies and resolving any conflicts. The full information on this stage is presented in "Section Effective permissions". The output from this stage is the set of all user resource permissions.
- **Stage 2: Extract Event Data.** This stage entails processing event log data sources to establish trust and sensitivity values without any prior knowledge. The process is detailed in "Section Trust and sensitivity", and the output is a heuristic score as to a subject's trust and also a resource's sensitivity.
- **Stage 3: Fuzzy Inference.** In this final stage, the outputs from Stage 1 and 2 are converted into linguistic representations through the fuzzification process. Next, the fuzzy inference engine is used to calculate fuzzy set membership, before the Mean of Maxima method is used to defuzzify set membership. The output of this stage is an updated list of permission with a risk score. The full details of the fuzzy process are in "Section Fuzzification".

In the remainder of this section, details are presented describing how access control permissions within NTFS are pre-processed before the fuzzy inference process can take place.

Effective permissions

As previously modelled, the effective permission set is defined as the user-permission-object (UPO) relationship $(u,p,o), \mu(u,p,o)$, where $u \in USERS$, $p \in PERMS$, and $o \in OBJECTS$. Furthermore, the fuzzy risk rating $\mu(u,p,o)$ is determined based on user trust, $\mu(u)$, permission power, $\mu(p)$, and resource sensitivity, $\mu(r)$.

It is necessary to convert the NTFS permission implementation into (u, p, o), accounting for group membership allocations, which are often utilised to implement role-based access control. It should be noted that the technique presented in this paper is generic, but the implemented permission policy will need to be translated into the effective permission model. To calculate the NTFS effective permission, we utilise the set of groups, $GROUPS = \{g_1, g_2, \ldots g_n\}$ and the following relationships:

- $GPO \subset GROUPS \times PERMS \times OBJECTS$ to represent the group-permission-object mapping of set items in the form of $(u,p,o), \mu(u,p,o)$ where $u \in USERS$, $p \in PERMS$, and $o \in OBJECTS$;

- $GG \subset GROUPS \times GROUPS$ to represent group-group mapping (g_1, g_2) where $g_1 \neq g_2$, $g_1 \in GROUPS$, and $g_2 \in GROUPS$;
- $UG \subset USERS \times GROUPS$ to represent user-group mapping (u, g), where $u \in USERS$, $g \in GROUPS$.

Using the above, the effective permissions can then be determined (*UPO*). Algorithm 1 describes the process of processing user-permission-object relations, *UPO*, and returning an updated version, accounting for permissions acquired through group associations. In a role-based approach, this would involve acquiring permissions that each user acquired through their role memberships. The algorithm takes as input the set of *USERS*, *PERMS*, *OBJECTS*, and *GROUPS* as well as their relationships *UPO*, *GPO*, *GG*, and *UG*.

The algorithm processes the users in turn (line 2), before processing each user-group relationship on line 4, searching for groups where the user has membership. In line 8 a recursive function (`proc(g, G)`) is then called to search for all the inherited group where each of the groups are a member. Finally, in line 9 all group-permission-object relationships ($(u, p, o) \notin UPO$) are processed to identify the permission level where a user is acquiring through group membership. A new permission entry is added into user-permission-object with permission and resource acquired from the group permission entry.

Trust and sensitivity

Once effective permissions have been calculated, it is then necessary to extract and set trust and sensitivity values. This could be done on a per resource and per user basis; however, this would be time-consuming for the user. It is often the case that resource permissions are set for groups of users and entire directory structures to minimise such effort. To reduce user effort, in this paper we make the following assumptions:

- A risk-averse approach is adopted whereby user trust is default to 'No Trust' and resource sensitivity is set to 'Unclassified';
- In the same way as permissions are allocated, sensitivity values are applied to a set of resource (e.g., directory structures), meaning that they are set on the parent resource and will automatically propagate through the resource inheritance hierarchy;
- User trust will be identified and assigned directly to individual users and not groups; and
- Conflict resolution is performed by the following two rules: (1) if multiple entries are identified, the lowest level of trust and highest level of sensitivity is assumed, and (2) explicit values always take priority over inherited values.

Algorithm 1 Algorithm to establish effective permissions

Input: Set of $USERS$, $PERMS$, $OBJECTS$, $GROUPS$
Input: Set of user-permission-object relationships, UPO
Input: Set of group-permission-object relationships, GPO
Input: Set of group-group relationships, GG
Input: Set of group-group relationships, GG
Output: Updated set of user-permission-object, UPO, reflecting the effective permissions

```
1:  procedure CalculateEffective
2:      for all u ∈ USERS do
3:          G = ∅
4:          for all (u_g, g) ∈ UG do
5:              if u = u_g then
6:                  G ∪ g
7:          for all g ∈ G do
8:              G ∪ GetAllGroups(g, U)
9:          for all (g, p, o) ∈ GPO do
10:             if G ∈ g then
11:                 if (u, p, o) ∉ UPO then
12:                     G ∪ (u, p, o)

13: procedure GetAllGroups(g, G)
14:     for all (g_1, g_2) ∈ GG do
15:         if g ∈ (g_1, g_2) then
16:             G ∪ g \ (g_1, g_2)
17:             G ∪ GetAllGroups(g, G)
```

In the presented solution, $\mu(u)$, $\mu(p)$, $\mu(o)$ values representing the user trust, permission power, and object sensitivity are extracted from the Microsoft NT environment. By default, the trust level is assigned to the lowest level ('No Trust') but is overwritten once the correct values are established.

In terms of extracting both user trust and resource sensitivity scores, the set of individual events (E) is processed. Specifically, to count the number of security related events that occur containing a user. Similarly, the set of events is processed to determine the number of times a resource occurs as a measure of sensitivity. Both approaches utilise the same technique where the event set, E, it literately processed, counting the number of times that either the user, u, or object, o, occur in the object list of each event (O) events of certain types. For example, when searching for a user, we count the number of times $u \in O$, iff the event id, I, is in a list of predefined events (IDs) of interest, $I \in IDs$.

In our experimental work on the Microsoft NTFS system, we search for user events that are deemed to be a negative activity. In terms of resource access, we count the occurrence of a resource being accessed. This is specifically monitored through the following event: `4663: An attempt was made to access an object`. Establishing user trust requires monitoring a greater number of events. In this specific application to Microsoft systems, we monitor all events deemed to have a 'High' or 'Medium' 'Potential Criticality' by Microsoft's own classification[1] and search specifically for the occurrence of user attribution. In total there are 88 event types to monitor. An example is `4724: An attempt was made to reset an account's password`.

Empirical analysis

In this section, systematic analysis is performed whereby previous research into detecting over-entitled permissions in Microsoft's NTFS is used as a case study. The target of Microsoft's NTFS results from the end-user motivation of this research, but it should be noted that the technique is transferable. The following research hypothesis is explored: analysing access controls using fuzzy logic, based on automatically extracted measures of user trust and resource sensitivity, enables easier detection of high-risk and potentially anomalous permission entries.

In this research, simulated file system and access control policies are created to enable a systematic comparison where ground truth knowledge is available. More specifically, knowledge on which access control entries are deemed as high-risk. It is necessary to develop and follow a process to create synthetic file systems and permissions allocations for the primary reason of having ground truth knowledge available as to each permission's risk. In other words, establishing a known benchmark that allows us to compute accuracy. Although real-world file systems can be used for analysis, manual analysis would be needed on each permission prior to analysis, which would be time-consuming and error prone. The method of using simulation (synthetic) data sets in access control research is common practice. Firstly, because ground-truth knowledge is often not available in real-world systems, and most significantly, organisations are unlikely going to share their access control policy with its known weaknesses. Of course, they could be anonymised, but organisations are often rightly security conscious and do not want to make their security data available. The method presented in this research is consistent with recent and key works that use a parameter-based approach to dataset generation, such as detecting anomalies in XACML policies (Aqib and Shaikh 2018), detecting and resolving anomalies while clustering ABAC Policies (El Hadj et al. 2018), and mining meaningful and rare roles from web application usage pattern (Gal-Oz et al. 2019). To be consistent with previous research, the results are examined by 30 final year undergraduate students, who have gained practical experience in undertaking analysis tasks, which includes reviewing access control policies. The large number is required to process the large number of simulated instances. In order to create realistic synthetic directory structures and allocation of permissions, the following parameters are changed:

- **Number of roles** is used to define the number of roles within the directory structure, which represent the number of organisational roles. For example, Management, Human Resources, etc.
- **Directory size** represents the depth and breadth of the synthetic directory structure. A directory structure will be created to the specified depth, with each directory containing the same number of subdirectories. For example, a directory size of 4 would result in the creation of a directory structure with a maximum depth of 4, and a breadth of 4 for each subdirectory. This exponential growth would create a directory size of $4^4 = 256$.
- **Total number of users** within the entire system would be equally distributed among the number of roles. For example, in a system with 100 users and 5 groups would result in 20 users per role.

[1] Microsoft's Best Practices for Securing Active Directory, Appendix L: Events to Monitor https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/appendix-l--events-to-monitor

Table 1 Experimental analysis parameter variation (minimum, maximum and stepsize)

Parameter	Minimum	Maximum	Stepsize
Number of roles	2	5	1
Directory Size	2	5	1
Total number of users	50	150	100
Percentage of over entitled users	2	10	4
User trust	0	100	50
Resource Sensitivity	0	100	50

- **Number of over-prescribed users** represents the number of users assigned permissions that are over entitled. Users, resource and permission level will be chosen at random.

The list generates the directory structures, allocating permissions, of which an increasing number are over-subscribed. The next aspects of creating the synthetic structures is to consider both user trust and resource sensitivity. As previously mentioned, these are defined as the number of times a user has been identified as being involved in adverse security actions, and resource sensitivity is identified as the number of times a specific resources is accessed. In this experimental methodology, both these parameters are increased in the following way:

- **User trust** will represent the normalised number of times a user is involved with security actions indicative of trying to violate a security policy. For example, an incorrect log-in (event ID: 4724), which as previously discussed in "Section Trust and sensitivity". As the test system is a Microsoft environment, we use the predefined list of security event entries and determine those that are of an adverse nature. The number of adverse security events will be incrementally increased. A low value is initially chosen to represent the number of adverse events for each user, and the number of events will be incrementally increased for over-prescribed users in order to represent a decreasing trust level.
- **Resource sensitivity** will represent the number of times a resource is accessed. A low value will be initially selected and applied across all resources to demonstrate a normal level of use. Resources used in instance of over-prescribed permissions will be incrementally increased to model a resource becoming more sensitive.

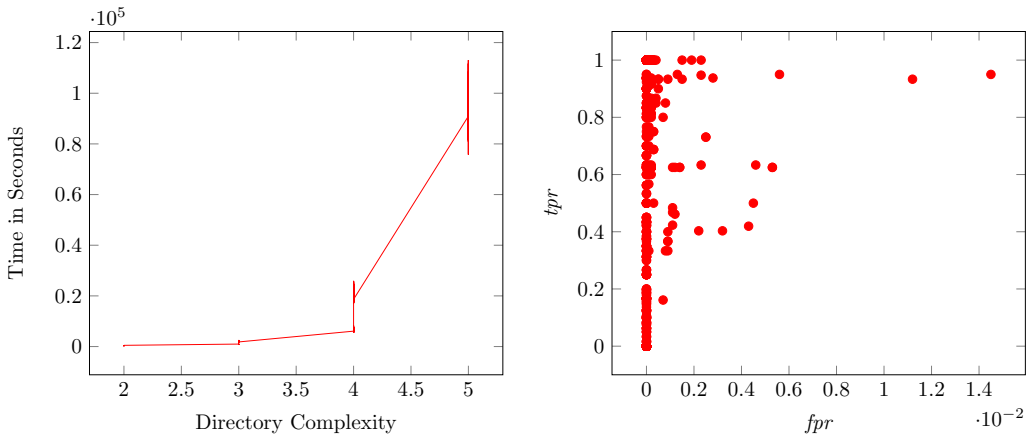
In terms of applying both user trust and sensitivity, the number of events introduced varied from 0 to 100, increasing in increments of 10. For the purpose of creating the datasets, the events are generated through a Powershell script to perform actions involving the user. For example, an incorrect authentication to trigger an incorrect login and resource access can be simulated through a user opening a directory.

Table 1 details the changing parameters in this incremental experiment. The number of roles is fixed to the number of levels of trust utilised in this research. In total there are 729 different system specifications to be used in this analysis. A Powershell script has been created to create the directory structure, create users and groups, and assign permissions. Furthermore, the script will also output the allocated user trust and resource sensitivity to be used for analysis purposes.

In this research, we evaluate the accuracy of the technique using the following measures:

1. True Positive Rate (*tpr*): the fraction of high-risk permissions correctly identified as being part of a high-risk permission.
2. False Positive Rate (*fpr = 1 - tnr*): the fraction of low to medium-risk permissions incorrectly identified as being high-risk.
3. True Negative Rate (*tnr*): the fraction of low to medium-risk permissions correctly identified as low to medium-risk;
4. False Negative Rate (*fnr = 1 - tpr*): the fraction of high-risk permissions incorrectly classified as low to medium-risk.
5. *Accuracy* is reported as the fraction of all samples correctly identified. More specifically, $Accuracy = \frac{tpr+tnr}{tpr+tnr+fpr+fnr}$.

It is worth noting that in this research we are particularly interested in establishing how many of the high-risk permissions are correctly classified, enabling the user to identify them and take any necessary mitigation. We are therefore focusing our evaluation on the true positive rate (*tpr*) and false negative rate (*fnr*). However, the *fpr* and *tnr* rates are still discussed. In relation to previous work, the motivation of this work is to help the end-user in identifying permissions of risk based on a fuzzy logic approach rather than a binary decision. It is, therefore, the case that establishing a confusion matrix purely on the classification of permissions as high-risk, i.e. treating it as a binary decision, is not appropriate, whereas focusing on the those both correctly classified as high-risk (*tpr*) and those incorrect missed (*fnr*) will enable a realistic understanding of how useful this tool is for the end-user.



(a) Average taken to process **(b)** Receiver Operator Curve

Fig. 6 Empirical analysis results

Table 2 Specifics of directory structures

Directory size	Directory size	Minimum number of permissions	Maximum number of permissions
2	4	715	2016
3	27	4425	12,326
4	256	38,138	106,237
5	3125	137,417	437,417

Results

In this subsection, the results of performing empirical analysis are presented and discussed. The analysis focuses on establishing both the performance and accuracy of the technique on directory structures, generated with different permission characteristics. Two illustrations are provided to help communicate the findings: the first is a Receiver Operator Curve (ROC) in Fig. 6b, showing the true positive rate (*tpr*) and false positive rate (*fpr*). This enables quick understanding of the trade-off between the technique correctly and incorrectly identifying permissions as high-risk. The second is Fig. 6a which presents the relationship between directory size and CPU time required to perform the analysis.

In Fig. 6a, it can be established that there is an exponential relationship between directory size and the time required to perform the analysis. For example, a directory size of 3, 4, 5 require on average a processing time of 24 min, 4 h, and 21 h. This lengthy-time period is due to the requirement to process each permission for each directory individually. It is worth noting that although this time is quite lengthy, the impact on the end-user is minimal as this analysis would be performed offline and a delay in the magnitude of hours is insignificant compared to the impact of the findings. Table 2 provides information on the directory structure, specifically the number of directories and permissions analysed. The information provided is for the instances with 0 over-prescribed permissions. The directory size details how many individual directories are included with each directory size. Both the minimum and maximum number of permissions within the variations is also presented, and it is evident that the number of permissions increases around 3 from 50 to 150 users. This is to be expected and demonstrates that the configuration of the test environments is as expected.

Figure 6b presents the ROC and it can be established that the results are in all instances better than chance (50/50). This is significant as it demonstrates that the technique can identify permissions of high-risk with a promising degree of accuracy. It can also be seen from Fig. 6b that usually, the *fpr* is 0 with varying *tpr*.

Table 3 provides the average accuracy values for the number of roles and directory size. Interestingly, it is evident that accuracy increases as the directory size increases. Furthermore, it can be seen in the table that the accuracy for a directory size of 2 and 3 are increasing with the number of roles. When considering the values contributing towards the accuracy, it is noticeable that the *tpr* is generally increasing with the number of roles, except for instances with 3 roles that demonstrate noticeably lower values. Overall the average *tpr* 0.53 and the *fnr* is 0.47. This demonstrates that there is a large portion of high-risk permissions that are incorrectly classified as normal. *fpr* and *tnr* are 0 and 1 for all instances, respectively. This illustrates the potential of the technique to correctly classified normal permissions as not high-risk

Table 3 Average accuracy values by role and directory size

Number of roles	Dir complexity	tpr	fpr	tnr	fnr	Accuracy
2	2	0.453	0.001	0.999	0.547	0.964
3	2	0.170	0.000	1.000	0.830	0.976
4	2	0.328	0.000	1.000	0.672	0.971
5	2	0.381	0.000	1.000	0.619	0.970
2	3	0.516	0.000	1.000	0.484	0.995
3	3	0.436	0.000	1.000	0.564	0.998
4	3	0.571	0.000	1.000	0.429	0.997
5	3	0.522	0.000	1.000	0.478	0.997
2	4	0.636	0.000	1.000	0.364	0.999
3	4	0.376	0.000	1.000	0.624	1.000
4	4	0.569	0.000	1.000	0.431	1.000
5	4	0.488	0.000	1.000	0.512	1.000
2	5	0.786	0.000	1.000	0.214	1.000
3	5	0.869	0.000	1.000	0.131	1.000
4	5	0.689	0.000	1.000	0.311	1.000
5	5	0.618	0.000	1.000	0.382	1.000
	Average	0.526	0.000	1.000	0.474	0.992

Table 4 Average accuracy values by number of over-prescribed users

Over-prescribed users	tpr	fpr	tnr	fnr	Accuracy
2	0.509	0.000	1.000	0.491	0.999
6	0.516	0.000	1.000	0.484	0.997
10	0.540	0.000	1.000	0.460	0.998
Average	0.521	0.000	1.000	0.479	0.998

(*tnr*), as well as not incorrectly classifying normal permissions as high-risk (*fpr*).

Table 4 presents the results based on the number of over-prescribed users. An observation can be made here that the *tpr* is increasing and the *fnr* is decreasing as the number of over-prescribed permission increase. This is because, as the number of introduced high-risk permissions increases, so does the number of high-risk permissions that are detected by the technique. A general observation here is that the change in accuracy values based on the increase introduced over-prescribed permissions is negligible, with a *tpr* increasing and a *fnr* decreasing by 0.031.

Table 5 presents average accuracy values based on the number of roles, varying trust, and varying sensitivity values. From the table, it is evident that accuracy values are gradually increasing along with the increase in trust and sensitivity values. Values with the highest accuracy values are generally those with lower trust and sensitivity values, which means that there are more events added in to represent decreasing user trust and an increase in resource sensitivity. More specifically, the *tpr* is increasingly along with the introduction of a higher number of events indicative of a user becoming less trustworthy and a resource becoming more sensitive. Although accuracy is always at the highest when both trust and sensitivity are highest, it is interesting to discover that accuracy values are greater when trust $= 100$ and sensitivity $= 0$ rather than trust $= 0$ and sensitivity $= 100$. However, after analysis of the raw results and file system permission allocation, it is evident that this reason is since other users are assigned to the resource with a high sensitivity, meaning that although the intended user is deemed to be of high trust, there are users with a lower trust level allocated to the resource, resulting in their permission's becoming higher risk.

In regard to the increasing number of roles, it is evident that as they increase the overall accuracy of the technique slightly decreases. This is because as the permissions are divided up into a greater number of roles with different permissions, it becomes harder to differentiate between permissions of high-risk as more permissions of a higher risk rating have been introduced.

When considering the *tpr*, *fpr*, *tnr*, and *tnr* with regards to an increase in the number of user trust and resource sensitivity events, the following points can be established.

The *tpr* values demonstrate a clear increasing pattern as both user trust decreases, and resource sensitivity increases. Furthermore, there is a pattern that the values decrease slightly as the number of roles increases. For example, with 2 roles, 100 trust and 100 sensitivity, the

Table 5 Average accuracy values by trust and sensitivity

Num of roles	Trust	Sensitivity	tpr	fpr	tnr	fnr	Accuracy
2	0	0	0.127	0.000	1.000	0.873	0.995
2	0	50	0.214	0.000	1.000	0.786	0.996
2	0	100	0.239	0.000	1.000	0.761	0.996
2	50	0	0.766	0.000	1.000	0.234	0.997
2	50	50	0.902	0.000	1.000	0.098	0.999
2	50	100	0.239	0.000	1.000	0.761	0.996
2	100	0	0.793	0.000	1.000	0.207	0.997
2	100	50	0.934	0.000	1.000	0.066	0.999
2	100	100	0.974	0.000	1.000	0.026	0.999
3	0	0	0.011	0.000	1.000	0.989	0.998
3	0	50	0.010	0.000	1.000	0.990	0.998
3	0	100	0.006	0.000	1.000	0.994	0.998
3	50	0	0.673	0.000	1.000	0.327	0.999
3	50	50	0.627	0.000	1.000	0.373	0.999
3	50	100	0.006	0.000	1.000	0.994	0.998
3	100	0	0.739	0.000	1.000	0.261	0.999
3	100	50	0.808	0.000	1.000	0.192	0.999
3	100	100	0.826	0.000	1.000	0.174	0.999
4	0	0	0.291	0.000	1.000	0.709	0.998
4	0	50	0.291	0.000	1.000	0.709	0.998
4	0	100	0.291	0.000	1.000	0.709	0.997
4	50	0	0.778	0.000	1.000	0.222	0.999
4	50	50	0.667	0.000	1.000	0.333	0.999
4	50	100	0.778	0.000	1.000	0.222	0.999
4	100	0	0.852	0.000	1.000	0.148	0.999
4	100	50	0.890	0.000	1.000	0.110	0.999
4	100	100	0.850	0.000	1.000	0.150	0.999
5	0	0	0.167	0.000	1.000	0.833	0.998
5	0	50	0.167	0.000	1.000	0.833	0.998
5	0	100	0.167	0.000	1.000	0.833	0.997
5	50	0	0.753	0.000	1.000	0.247	0.999
5	50	50	0.786	0.000	1.000	0.214	0.999
5	50	100	0.753	0.000	1.000	0.247	0.999
5	100	0	0.793	0.000	1.000	0.207	0.999
5	100	50	0.825	0.000	1.000	0.175	0.999
5	100	100	0.848	0.000	1.000	0.152	0.999

tpr is 0.97 and for the same values with 5 roles, the *tpr* decreases to 0.85. As previously noted, this is is due to the increasing variation in allocated permissions, meaning that it becomes harder to differentiate high-risk permissions.

The *fpr* vales are consistently 0 for all datasets, establishing that no instances of normal permissions are incorrectly classified as high-risk. Similarly, the *tnr* values are consistently 1 demonstrating the technique's capability to correctly identify non-high-risk permissions as normal. This is significant as it enables the end-user to not waste time investigating false positives.

The *fnr* is not as good as the measures previously discussed. The number of high-risk permissions incorrectly identified as normal is greater than 0 throughout all instances. The *fnr* is highest for instances containing users of a higher trust level and lower resource sensitivity, which results in it being more difficult to identify high-risk permissions. It is, however, important to note that the *fnr* values are decreasing as the number of less trustworthy users and

resources with higher sensitivity are increasing. For example, with a role count of 2 and 0 for both trust and sensitivity values, the *fnr* is 0.76. This value decreases to 0.03 when increasing trust and sensitivity to 100. It is also interesting to note that the *fnr* increases slightly as the number of roles increases. More specifically, the average *fnr* is 0.03 with a role number of 2, trust of 100 and sensitivity of 100. The values increase to 0.15 for the simulated directory structure with a role count of 5 but the same trust and sensitivity parameters.

Conclusion

Analysing access control systems is a common activity for those wishing to review and improve access control implementations. However, it was established that binary classification mechanisms have difficulty when classifying permissions as normal or irregular. In this work, we pursued the hypothesis of analysing access controls using fuzzy logic, based on automatically extracted measures of user trust and resource sensitivity, enables easier detection of high-risk and potentially anomalous permission entries. This results in the modelling of file system permissions, taking into considering user trust and resource sensitivity. A practical implementation was then developed to gain empirical observations. In developing this mechanism, it was necessary to devise a way to establish user trust and resource sensitivity without the user needing to provide additional information. The identified method was through monitoring adverse security actions and interaction with underlying resources in event logs.

The technique presented in this paper was tested and resulted in an overall average accuracy of 99%. This demonstrates the potential of the technique, especially when considering that binary techniques struggle to gain beyond 90% accuracy. This demonstrates the potential of the technique and supports the hypothesis; however, it should be noted that it does add a dependency on event data to establish user trust and resource sensitivity, which in some system may be incomplete or not available. In future work, research will be performed into how further information sources can be combined into the trust and sensitivity measures.

Acknowledgements
A special thanks to Janine Hamilton, Tony Dove, and Ian Sharp for their support throughout the project, under the UK's Digital Catapult Centre under its Researcher in Residency Fellowship Programme

Authors' contributions
SP was responsible for the development of the technique and drafting the manuscript. SK contributed towards running the experiments, interpreting the findings, and drafting the manuscript. All authors read and approved the final manuscript.

Funding
This work was undertaken during a project funded by the UK's Digital Catapult Researcher in Residency Fellowship programme (Grant Ref: EP/M029263/1). The funding supported the research, development, and empirical testing presented in this paper.

Availability of data and materials
All experimental datasets, scripts and software are available from the corresponding author upon request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 14 July 2021 Accepted: 17 January 2022
Published online: 02 March 2022

References

Abie H, Balasingham I (2012) Risk-based adaptive security for smart IoT in ehealth. In: Proceedings of the 7th international conference on body area networks. ICST (Institute for Computer Sciences, Social-Informatics and...), pp 269–275

Ahmed A, Alnajem A (2012) Trust-aware access control: how recent is your transaction history? In: 2012 second international conference on digital information and communication technology and it's applications (DICTAP). IEEE, pp 208–213

Aqib M, Shaikh RA (2018) A tool for access control policy validation. J Internet Technol 19(1):157–166

Atlam HF, Alenezi A, Walters RJ, Wills GB, Daniel J (2017) Developing an adaptive risk-based access control model for the internet of things. In: 2017 IEEE international conference on Internet of Things (iThings) and IEEE green computing and communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and Ieee Smart Data (SmartData). IEEE, pp 655–661

Atlam HF, Walters RJ, Wills GB, Daniel J (2019) Fuzzy logic with expert judgment to implement an adaptive risk-based access control model for IoT. Mobile Networks and Applications, pp 1–13

Bernabe JB, Ramos JLH, Gomez AFS (2016) Taciot: multidimensional trust-aware access control system for the internet of things. Soft Comput 20(5):1763–1779

Bodnar T, Tucker C, Hopkinson K, Bilén SG (2014) Increasing the veracity of event detection on social media networks through user trust modeling. In: 2014 IEEE international conference on big data (Big Data). IEEE, pp 636–643

Cheng P-C, Rohatgi P, Keser C, Karger PA, Wagner GM, Reninger AS (2007) Fuzzy multi-level security: an experiment on quantified risk-adaptive access control. In: IEEE symposium on security and privacy. SP'07. IEEE, pp 222–230

Cheng P-C, Koved L, Singh KK (2016) Trust/value/risk-based access control policy. Google Patents. US Patent 9,432,375

El Hadj MA, Khoumsi A, Benkaouz Y, Erradi M (2018) Formal approach to detect and resolve anomalies while clustering abac policies. EAI Endorsed Transactions on Security and Safety 5(16)

Ferraiolo D, Kuhn DR, Chandramouli R (2003) Role-based access control. In: In Proceedings of the NIST-NSA National (USA) computer security conference, pp 554–563

Folorunso O, Mustapha OA (2015) A fuzzy expert system to trust-based access control in crowdsourcing environments. Appl Comput Inform 11(2):116–129

Friedlob GT, Schleifer LL (1999) Fuzzy logic: application for audit risk and uncertainty. Manag Audit J 14(3):127–137

Gaddam A, Aissi S, Kgil T (2014) Data sensitivity based authentication and authorization. Google Patents. US Patent App. 14/303,461

Gal-Oz N, Gonen Y, Gudes E (2019) Mining meaningful and rare roles from web application usage patterns. Comput Secur 82:296–313

Helil N, Halik A, Rahman K (2017) Non-zero-sum cooperative access control game model with user trust and permission risk. Appl Math Comput 307:299–310

Hu H, Ahn G-J, Kulkarni K (2013) Discovery and resolution of anomalies in web access control policies. IEEE Trans Dependable Secure Comput 10(6):341–354

Khan S, Parkinson S (2018) Eliciting and utilising knowledge for security event log analysis: an association rule mining and automated planning approach. Expert Syst Appl 113:116–127
Khan S, Parkinson S (2019) Discovering and utilising expert knowledge from security event logs. J Inf Secur Appl 48:102375
Kiedrowicz M, Stanik J, Kubiak B, Maślankowski J (2015) Selected aspects of risk management in respect of security of the document lifecycle management system with multiple levels of sensitivity. In: Kubiak BF, Maślankowski J (eds) Information management in practice, pp 231–249
Kozhakhmet K, Bortsova G, Inoue A, Atymtayeva L (2012) Expert system for security audit using fuzzy logic. In: Midwest artificial intelligence and cognitive science conference, p 146
Leichtenstern K, André E, Kurdyukova E (2010) Managing user trust for self-adaptive ubiquitous computing systems. In: Proceedings of the 8th international conference on advances in mobile computing and multimedia, pp 409–414
Li N, Tripunitara MV (2005) On safety in discretionary access control. In: 2005 IEEE symposium on security and privacy (S&P'05). IEEE, pp 96–109
Lu X, Qu Z, Li Q, Hui P (2015) Privacy information security classification for internet of things based on internet data. Int J Distrib Sens Netw 11(8):932941
Mahalle PN, Thakr, PA, Prasad NR, Prasad R (2013) A fuzzy approach to trust based access control in internet of things. In: Wireless VITAE 2013. IEEE, pp 1–5
McLeod S (2007) Maslow's hierarchy of needs. Simply Psychol 1
Mhetre NA, Deshpande AV, Mahalle PN (2016) Trust management model based on fuzzy approach for ubiquitous computing. Int J Ambient Comput Intell (IJACI) 7(2):33–46
Nekooei SM, Chen G, Rayudu RK (2017) Automatic design of fuzzy logic controllers for medium access control in wireless body area networks-an evolutionary approach. Appl Soft Comput 56:245–261
Ni Q, Bertino E, Lobo J (2010) Risk-based access control systems built on fuzzy inferences. In: Proceedings of the 5th ACM symposium on information, computer and communications security. ACM, pp 250–260
Osborn S, Sandhu R, Munawer Q (2000) Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Trans Inf Syst Secur (TISSEC) 3(2):85–106
Ouaddah A, Mousannif H, Elkalam AA, Ouahman AA (2017) Access control in the internet of things: big challenges and new opportunities. Comput Netw 112:237–262
Park Y, Gates SC, Teiken W, Cheng P-C (2011) An experimental study on the measurement of data sensitivity. In: Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, pp 70–77
Park Y, Gates C, Gates SC (2013) Estimating asset sensitivity by profiling users. In: European symposium on research in computer security. Springer, pp 94–110
Park Y, Teiken W, Rao JR, Chari S (2016) Data classification and sensitivity estimation for critical asset discovery. IBM J Res Dev 60(4):2–1
Parkinson S (2017) Use of access control to minimise ransomware impact. Netw Secur 2017(7):5–8
Parkinson S, Crampton A (2016) Identification of irregularities and allocation suggestion of relative file system permissions. J Inf Secur Appl 30:27–39. https://doi.org/10.1016/j.jisa.2016.04.004
Parkinson S, Khan S (2018) Identifying irregularities in security event logs through an object-based chi-squared test of independence. J Inf Secur Appl 40:52–62
Parkinson S, Somaraki V, Ward R (2016) Auditing file system permissions using association rule mining. Expert Syst Appl 55:274–283. https://doi.org/10.1016/j.eswa.2016.02.027
Parkinson S, Vallati M, Crampton A, Sohrabi S (2018) Graphbad: a general technique for anomaly detection in security information and event management. Concurr Comput Pract Exp 30(16):4433
Parkinson S, Khan S, Bray J, Shreef D (2019) Creeper: a tool for detecting permission creep in file system access controls. Cybersecurity 2(1):14
Patyra MJ, Mlynek DJ (2012) Fuzzy logic: implementation and applications. Springer, Berlin
Pfleeger CP, Pfleeger SL (2002) Security in Computing. Prentice Hall Professional Technical Reference
Rahmati A, Fernandes E, Eykholt K, Prakash A (2018) Tyche: a risk-based permission model for smart homes. In: 2018 IEEE cybersecurity development (SecDev), pp. 29–36. IEEE
Ray I, Kumar M (2006) Towards a location-based mandatory access control model. Comput Secur 25(1):36–44
Ryutov T, Zhou L, Neuman C, Leithead T, Seamons KE (2005) Adaptive trust negotiation and access control. In: Proceedings of the tenth ACM symposium on access control models and technologies, pp. 139–146. ACM
Salem MB, Bhatti R, Solderitsch J (2013) Method and system for resource management based on adaptive risk-based access controls. Google Patents. US Patent App. 13/774,356
Samarati P, de Vimercati SC (2000) Access control: policies, models, and mechanisms. In: International school on foundations of security analysis and design, pp. 137–196. Springer, Berlin
Sandhu RS, Samarati P (1994) Access control: principle and practice. IEEE Commun Mag 32(9):40–48
Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47
Shahriar H, Zulkernine M (2011) A fuzzy logic-based buffer overflow vulnerability auditor. In: 2011 IEEE ninth international conference on dependable, autonomic and secure computing. IEEE, pp 137–144
Shaikh R, Sasikumar M (2015) Data classification for achieving security in cloud computing. Procedia Comput Sci 45:493–498
Sherwin K (2016) Hierarchy of trust: the 5 experiential levels of commitment. https://www.nngroup.com/articles/commitment-levels
Stanik J (2017) System risk model of the it system supporting the processing of documents at different levels of sensitivity. In: MATEC Web of Conferences, vol. 125, p. 02011. EDP Sciences
Younis YA, Kifayat K, Merabti M (2014) An access control model for cloud computing. J Inf Secur Appl 19(1):45–60
Zhou A, Li J, Sun Q, Fan C, Lei T, Yang F (2015) A security authentication method based on trust evaluation in VANETs. EURASIP J Wirel Commun Netw 2015(1):1–8

Publisher's Note