Towards A Multi-Tiered Knowledge-Based System for Autonomous Cloud Security Auditing

Saad Khan, Simon Parkinson
Department for Informatics, School of Computing and Engineering, University of Huddersfield, HD1 3DH
<firstname>.<surname>@hud.ac.uk

Abstract
Every cloud platform has a large number of software components, making it difficult to manage the security of the entire system. This paper discusses the requirement for an intelligent cloud security auditing solution, and an expert system architecture is presented. The solution can identify data confidentiality threats in the OpenStack cloud platform, as well as propose solutions to remove vulnerabilities before an attack occurs. Data confidentiality threats cover a wide range of security risks where attackers usually try to steal/corrupt personal data and are a major concern of users. For this reason, cloud infrastructures need frequent security auditing. The key features of the proposed expert system architecture include: acquisition of information detailing the latest cloud security threats and solutions, the conversion of acquired raw data into usable format, the application of a forward chaining inference algorithm, and the ability for the user to add/modify knowledge, which is then utilised to provide feasible solutions in ranked order. These components provide an automated mechanism to generate human-readable audit reports, improving the overall security status without the need for expert knowledge.

Introduction
Cloud infrastructures are a very lucrative platform for attackers as they contain a large amount of personalized data and information. Personal data can be of significant interest for an adversary, and different data sources can have different privacy implications. For example, acquiring data regarding a website’s finance transactions would contain an individual’s buying habits, whereas personal data on a social media platform may be sufficiently detailed to enable identity theft and social-engineering attacks. In all cases, the breach of data is harmful for users, and any damaging news would be financially challenging for the service provider. A fundamental of data protection is through enforcing the Confidentiality, Integrity and Availability (CIA) model (Olivier 2002).

This paper focuses on the confidentiality aspects, which involve protecting the privacy and secrecy of data. In order to develop a model of cloud security, it is necessary to develop a cloud platform for experimental analysis. OpenStack is chosen for its large market-share, largely due to its performance and other benefits, discussed in: (Steinmetz, Perrault et al. 2012, Rahmani, Sundararajan et al. 2013).

When a user uploads data to a cloud service, the data is essentially out of user control. When using services provided by Google drive, Dropbox and Microsoft OneDrive, etc., user do not even know the physical location of data (Clarke 2013) and they are trusting the service provider to have implemented adequate security provisions. Weaknesses like unencrypted data, poor authentication schemes, weak passwords, unmanaged access control, virtualization security issues and targeted attacks make it difficult to protect the secrecy, privacy and confidentiality of data. Due to the sensitive nature of cloud platform, and the continuous rise of security threats, such cloud security auditing solutions are required that can proactively determine the threats, and help to eradicating them in advance, aiding to improve CIA in regards to the cloud infrastructure.

Why use expert system to increase cloud security?
The first expert systems was developed during 1960s by Edward Feigenbaum. Since then, there has been a large volume of work in developing the expert system paradigm and in applying them to many real-world problems. Expert systems have been used in many security applications; however, to the best of the author’s knowledge, there is an absence of an expert system for cloud security auditing, specifically targeting data confidentiality.

A cloud platform consists of many smaller components such as computation, networking, storage etc. modules (Sefraoui, Aissaoui et al. 2012). Even if a single component is vulnerable to internal or external threats, it has the potential to make the entire cloud infrastructure vulnerable, including any sensitive, in-use and critical user data. Expert systems can play a vital role in the security auditing of cloud, and the foremost reason is due to the large requirement on maintaining the skills of a human analyst to identi-
Related work

This section reviews the existing work in the field of knowledge/rule-based expert systems. It determines the architecture, components, implementation and evaluation techniques of both cyber-security and other successful expert systems applications in different domains.

Components of Expert Systems

Liao conducted a literature survey from 1995 to 2004 and identified 11 types of expert systems such as rule-based system, knowledge-based system etc (Liao 2005). During their studies, they concluded that rule-based expert systems were the most widely used type. According to (Hatzilygeroudis and Prentzas 2001), there are four major components of expert system (1) Knowledge Base: set of rules and facts, (2) Database of Facts: information gathered about the current system using manual and/or automated inputs, (3) Inference Engine: deduction system to draw conclusion and (4) Explanation Mechanism: justifies the provided solutions. The system combines the rules with given facts to solve problems and provide justified explanation. A recent study proposes the following two steps process to create a knowledge base: (1) knowledge gathering from literature review and human expert opinion and (2) knowledge representation in the form of conditional rules (Rajsiri, Lorré et al. 2010). The rules can be created by IF-THEN statement or AND/OR trees (Novak and Oreški 2015). Inference engines usually implement two kinds of algorithms: forward and backward chaining. A recent paper presented performance comparison of both algorithms (Kamley, Jaloree et al. 2016). They claim that forward chaining deduce conclusions from the given facts while backward chaining draws facts from the given conclusions. Forward chaining explores every possibility while resolving rules, which leads towards an exhaustive exploration of the inference state-space. However, their discussion of backward chaining describes how they can avoid inference of unnecessary clauses as it is goal-driven.

Existing Cyber-Security Expert Systems

The Open Security Knowledge Engineered (OpenSKE) framework (Gamal, Hasan et al. 2011) conducts security analysis based on system information and a provided knowledge base of vulnerabilities. They have used online community-driven databases, such as Common Vulnerabilities and Exposures Enumeration (CVE) for decision algorithm to implement their logic. Chang and Lee developed an expert system using ISO/IEC 27001 and ISO/IEC 27005:2008 standards (Chang and Lee 2013). They implemented fuzzy if-then rules as their inference engine. The important feature of their paper is the overall risk evaluation of system. Miller et al. used Weighted and Order Weighted Averages with Evolutionary Algorithms to perform cyber-security assessment (Miller, Wagner et al. 2016). They also performed overall evaluation by expert rating of system components.

Liu and Fang improved the efficiency of Intrusion Detection Systems (IDS) by eliminating false positives with an expert system (Liu, Fang et al. 2011). Cloud infrastructures have a large number of components, applications and users, which make it difficult to analyze each packet. They have used a combination of Certainty Factor and fuzzy rules to monitor and aggregate the security alerts. One author developed a Fuzzy Rule Based Expert System for Cyber Security (FRBCS) system to identify cyber threats (Goztepe 2012). It contains user interface, forward chaining inference engine, attacks information and 83 fuzzy if-then rules. A framework named Cyber Security Modelling Language (CySeMoL) was developed for determining the chance of successful attacks against system (Holm, Sommestad et al. 2013). The framework has 22 classes with 102 attributes and 54 solutions, which are used for Attack-Path Generation and Assessment. The important feature of this system is the validation and verification of proposed solutions. Neuro-Fuzzy and Mamdani inference fuzzy system are also being used in expert systems to evaluate and quantify trust in B2C E-commerce websites (Nilashi, Bagherifard et al. 2011). They built decision matrices for rule-base, assigned weight to each item and determined consistency index/ratio with the opinion of 10
experts and 150 students on Likert scale. Another expert system based on fuzzy logic, was developed by (Kozhakhmet, Bortsova et al. 2012), which categorizes threats as low/moderate/high. Various factors such as authentication, monitoring, logs and password management in terms of data security are considered. A design of rule-based expert system for automated security auditing is presented by (Tsudik and Summers 1990). Their system has four steps: occurrence, detection, follow-up and archiving. Each rule contains violation type, users, location, time and threshold. When any violation rule is triggered, it takes action accordingly and archives the results for future. SecureScan\footnote{Information on SecureScan ad BeyondSaas are available at http://www.tripwire.com/products-and-needs/ https://www.beyondtrust.com/products/beyondsaas/} is a commercial cloud security product, which determines cloud infrastructure vulnerabilities. It informs about severity level, vulnerability name, ports, impact and solution. BeyondSaas is a fully featured vulnerability scanner for cloud. It performs auditing, vulnerability scans, remediation techniques, SaaS application crawling and performs the scan in secure manner.

Based on the above literature review, there are expert systems which assist in suggesting accurate security solutions against complex and wide range of attacks on cloud platform; however, the significant limitations is that the tools require expert knowledge to utilize correctly and are often focused on a targeted sub-category of security analysis. In this paper, a prototype architecture is discussed which can: look for vulnerabilities in an automated manner, propose correct and applicable solutions, provide user the ability to increase knowledge of expert system and present the results in human-readable format.

**Other Expert System Applications**

In one related work, the authors developed an expert system to solve construction related problems (Mosa, Rahmat et al. 2013). The knowledge was acquired from both literature review and human experts. Classification of knowledge was done using directed graphs and decision matrix, which also contains \textit{IF} and \textit{THEN} segments. As medical diagnosis is knowledge intensive, expert systems have been developed to diagnose vitamin deficiencies in humans using forward chaining algorithm (Novaliendry, Yang et al. 2015). The authors created a list of symptoms for several vitamin deficiencies and Decision Tree to extract \textit{if-then} rules. Forward chaining algorithm and Bayes Theorem can also be combined together for developing expert system to diagnose teeth and oral diseases in children (Maharani, Degen et al. 2015). Similarly, (Windriyani, Kom et al. 2013) proposed an expert system using forward chaining algorithm and interviews for eliciting knowledge to detect mental disorders. They first created Decision Table, Decision Tree and finally \textit{if-then} rules for the inference engine.

**Expert System Architecture for auditing cloud data security**

Considering the aforementioned discussion, we propose a novel architecture for an expert system, which can aid the proactive elimination of data confidentiality threats in cloud systems. The expert system is aimed to be used by OpenStack cloud administrators and service providers. It will help users with little or no expert knowledge of auditing to identify threats and implement respective solutions. The system has eleven components and two knowledge-bases, which are defined in the following and shown in Figure 1. A case study example of the proposed architecture is provided later in the paper.

**Component 1: Identification of resources for auditing**

As mentioned earlier, cloud infrastructures have a large number of components and we need to identify the ones which can be used to exploit data confidentiality. After carefully analyzing many survey papers (Subashini and Kaviitha 2011, Chen and Zhao 2012, Modi, Patel et al. 2013, Fernandes, Soares et al. 2014, Ab Rahman and Choo 2015, Hussein, Khalid et al. 2016), we concluded that following are of key concerns (threat categories):

- Authentication Issues (AI);
- Drive-By Attacks (DBA);
- Encrypting Data and Communication (EDC);
- Hypervisor Attacks (HA);
- Inefficient and Insufficient Monitoring (IIM);
- Insecure Services and Applications (ISA);
- Insider Threat (IT);
- Malicious VMs (MV);
- Malware Infection (MI);
- Multi-Tenancy Issues (MTI); and
- Ransomware (R).

These key areas of concern have been highlighted as major concerns when protecting secrecy of data in cloud. However, due to the flexibility of the proposed architecture, we can later add more categories of issues such improper patch management, misconfiguration etc.

**Component 2: Manual and Automated Cloud reconnaissance**

The purpose of this component is to create a Database of Facts. Its purpose is to be used alongside the current state of underlying cloud infrastructure to use when performing automated penetrating testing, with facts acquired from both acquisition software and manual inputs. This process will specifically target selected resources and output data
in a raw, condensed format. For the manual input, there will be a user interface with multiple-choice questions, where the administrator will select the most relevant option. The questions will only ask about those security features which cannot be determined automatically, such as the details of a used encryption algorithm. The automated testing will collect most of the information using tools like Nmap, Metasploit, sqlmap, Burp Suite etc.

Component 3: Converting raw data into useful information
Before starting the conversion process, we first need to define a complete set of attributes, A, from all threat categories, which should reflect the output of the cloud reconnaissance phase in a structured manner. These attributes will also help create the rule-base. Formally:

\[ A = \{A_{\text{DBA}}, A_{\text{EDC}}, A_{\text{HAA}}, A_{\text{HIM}}, A_{\text{RTI}}, A_{\text{MTI}}, A_{\text{MTF}}, A_{\text{R}}\} \]

where \( A_x = \{a_1, a_2, a_3 \ldots a_n\}, X \) is the name of all categories and \( a_k \) represents single attribute.

Component 4: Defining Universal Set for threats and solutions
To identify threats and propose solutions for a given system, we must have a complete set of threats and solutions to choose from. As it is very difficult to gather knowledge about all attacks and security solutions, the knowledge acquisition process is one of the major bottlenecks of proposed systems. In addition, this process might also generate subjective knowledge while searching literature online or taking human expert opinion.

Currently, all threats and solutions are gathered from a literature review by searching respective keywords in reputable outlets such as IEEE Xplore Digital Library, ACM Digital Library, ScienceDirect, Springer and Elsevier which have a systematic peer review process. Assume that \( U_T \) represents threats and \( S \) represents solutions, then the universal set \( U_T \) contains all threats and \( U_S \) contains all solutions within the scope of system. Both \( U_T \) and \( U_S \) are the separate knowledge-bases of the expert system. In addition, TS contains the identified threats in cloud by triggering condition/IF-part of rules in \( U_T \) and \( S \) represent the proposed solutions by expert system, whose elements are essentially the consequence/THEN-part of triggered rules in \( U_S \). Formally:

\[ U_T = \{T_1, T_2, T_3, T_4 \ldots T_N\}, \text{where } T_i \text{ is (IF-THEN) rule} \]
\[ U_S = \{S_1, S_2, S_3, S_4 \ldots S_N\}, \text{where } S_i \text{ is (IF-THEN) rule} \]

Also, \( TS \subseteq U_T \) and \( SS \subseteq U_S \)

Every threat and solution will be assigned a weight, using Likert scale (1-5), depending on the evaluation of its publisher, \( W_{U_T} \) and \( W_{U_S} \) respectively. The weight of each threat and solution will be used for ranking expert system output and overall security status of cloud. Formally:

\[ W_{U_T} = \{W_{T_1}, W_{T_2}, W_{T_3}, W_{T_4} \ldots W_{T_N}\} \]
\[ W_{U_S} = \{W_{S_1}, W_{S_2}, W_{S_3}, W_{S_4} \ldots W_{S_N}\} \]

Where \( 1 \leq W_{\{T_i, S_i\}} \leq 5 \) and \( 1=\text{low and } 5=\text{high} \)

Component 5: Adding knowledge into \( U_T \) and \( U_S \)
Another important feature that distinguishes this expert system is the provision of adding knowledge into universal sets. The user will be presented an interface for adding new rules. With a list of all attributes from set \( A \), if-statement, then-condition and input/output parameters, the user can generate new rule or modify/delete exiting ones. This provides flexibility in the system to enable the expert system to be extended in the future to accommodate new threat and solution combinations.

Component 6: Extracting TS from attribute/value pairs using \( U_T \)
This task will be performed by an intelligent inference algorithm, which is also able to process large quantities of data. As evident from the related work, a forward chaining algorithm is a feasible solution for this inference engine because we need to extract all possible threats. The algorithm must identify all threats, which exist in cloud systems where their information is present in knowledge-base.

Component 7: Extracting SS from TS using \( U_S \)
This task will also be performed by an intelligent inference algorithm, which can suggest a minimal set of relevant solutions, such that one solution can mitigate one or more threats. Again, to complete this task, intelligent forward chaining algorithm is the feasible one. This component also needs to handle resolve potential conflicts whereby a suggested solution may solve or create new threats.

Component 8: Verification and Validation of SS
The purpose of this component is to determine whether the solutions in SS are correct and applicable according to the given cloud platform. The correctness factor can be determined and justified with the explanation mechanism i.e. explanation and step-by-step procedure of how expert system reached its decision. The applicability factor requires human interaction, where user will be asked whether they can deploy the given solution. If the solution is not viable, then a new solution will be presented, and new knowledge stored about the incompatibility.

---

2 In this architecture we only consider open source security penetration tools.
Component 9: Automatically identifying the next feasible solution

W_{UT} and W_{US} contain a number from 1 to 5 on Likert scale, which depicts the ranking of each threat and solution. The first priority of expert system will be to output solutions with the maximum weight. If the solution is not accepted by the user, the expert system will output solution with second maximum weight and so on. In this way, expert system can provide flexible and most relevant solutions within the scope of its knowledge.

Component 10: Heuristics Engine

It is one of the integral components of expert system and has two major functions. First, learn and save the results of expert system, every time it is executed. If same or similar set of attribute/value pairs are received again from cloud reconnaissance, the expert system will display the previously calculated results. Secondly, if user or penetrating testing fails to provide required input, the expert system will corroborate with the heuristic engine to estimate or predict missing values. Hence, instead of an error, the expert system will render the best solution in given environment.

Component 11: Report and overall cloud security evaluation

The purpose of this component is to generate human-readable reports for both IT specialists and users with limited knowledge, to the same high standard of a human security expert. In addition, the report will also generate the overall ratings of system by taking average of $W_{TS}$ elements and categorizing it into three severity levels: major security flaws (3-5), minor security flaws (1-3) and recommendations (0-1) for improving the overall health of system.

Case-Study example

This section presents a case-study example for demonstrating the workings of the proposed expert system. Although, cloud systems inherit many security flaws from traditional systems, there are some vulnerabilities that are more common in cloud systems. One such example is Virtual Machine (VM) escaping attack (under the category of Hypervisor Attacks) is explained below.

VM Escaping attack

The knowledge to identify and prevent this attack is acquired from publication by (Perez-Botero, Szefr et al. 2013). The reader should consult this paper to find further information explaining the meaning of the presented attributes and values. Only one publication is considered for demonstration purpose, hence the results do not present all of the threat attributes and solutions for an attack of this type. A VM escaping attack is where the attack is able to break outside of the VM and gain access/control of the host operating system. In a cloud system, many virtual machines will be running in order to provide all the required services. An attack of this type has significant implications as the attacker may be able to gain unforeseen access to user’s data, and thus has significant implications for user privacy.

Assume, we have following elements in attribute set, universal threat/solution sets and their respective weights:

$$\begin{align*}
A &= \{\text{Physical CPU (PC), Virtual CPU (VC), Current Privilege Level (CPL), Symmetric Multiprocessing (SMP), Hypervisor Add-ons (HA)}\} ; \\
U_T &= \{\text{If (VC != mirror(PC)) then CVE-2010-4525, If (CPL == Ring-3 for guest) then ‘SMP is malicious’, If (HA == true) then CVE-2008-3687}\}; \\
W_{UT} &= \{3, 4, 3\}; //\text{severity of threat} \\
U_S &= \{\text{If(CVE-2010-4525) then ‘update Linux kernel’, If(SMP is malicious) then ‘update KVM hypervisor’, If(CVE-2008-3687) then ‘remove buffer overflow in XSM:FLASK by updating’}\}; \\
W_{US} &= \{5, 5, 5\}; //\text{efficiency of solution}
\end{align*}$$

The first step of the expert system is cloud reconnaissance, which will gather system properties using Virsh tool\(^3\) in a Kernal-based Virtual Machine (KVM). It will output a raw text file with a lot superfluous data. The next phase, converting raw data into useful information, will extract the relevant information from the text file and populate values of all attributes in set $A$. At this stage, some tools like Nmap have ability to output in XML format, which makes it quite easier to parse. For other tools, we have to create individual output parsers, in accordance with their output-format. As attribute/value pair influence final results, Heuristic Engine can also help to fill in the gaps. Currently, we only have very small system with no dependent rules, which is why both inference algorithms in extracting TS from attribute/value pairs using UT and extracting SS from TS using US phases will conduct one-to-one mapping and output as follows:

$$\begin{align*}
TS &= \{ \text{CVE-2010-4525, ‘SMP is malicious’, CVE-2008-3687}\}; \\
SS &= \{\text{‘update Linux kernel’, ‘update KVM hypervisor’, ‘remove buffer overflow in XSM:FLASK by updating’}\};
\end{align*}$$

The expert system will justify its results in verification and validation of SS phase. Essentially, it will display the

\(^3\) Further information on Virsh tool is available at: https://linux.die.net/man/1/virsh
sequence of rule triggering. For example, consider first rule in $U_1$. It will display that virtual CPUs are not mirroring the behavior of physical CPU, hence there exists CVE-2010-4525 vulnerability. As there are no other solutions for the same problem, if user rejects the given solution, the system will exit in error state. If user accepts the solution, (1) the results will be saved in Heuristic Engine and (2) the system will output a report and overall status of cloud as following:

<table>
<thead>
<tr>
<th>Average of $W_{TS}$ = (3+4+3)/3 = 3.33. The system has major security flaws as the value is between 3 and 5.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The format of audit report is as follows:</td>
</tr>
<tr>
<td>1. List of all identified threats (TS) along with their Solutions (SS)</td>
</tr>
<tr>
<td>2. The severity of individual threat and efficiency of each solution in term of weights</td>
</tr>
<tr>
<td>3. The overall security status of cloud</td>
</tr>
<tr>
<td>4. Time elapsed</td>
</tr>
</tbody>
</table>

**Evaluation**

Notice that the system’s productivity and performance depends on the amount and quality of knowledge acquisition process. Furthermore, in some cases, the expert system relies on the third party tools to collect information about the cloud system. We are unable to measure the validity and reliability of third party tools, especially under different kinds of cloud setups. Finally, there is no internal solution verification process i.e. there is no context awareness in proposed system.

The benefit of proposed expert system is that it can perform authentic and complete auditing process. By allocating a separate VM in cloud for this system, administrator can regularly manage the security with minimal effect, without interrupting normal operations. Furthermore, as every cloud has different requirements, administrator can add custom rules or modify existing ones. And with the help of heuristic engine, the expert system will adapt and learn from its each execution. The audit report and overall security rating can provide holistic health view of cloud to everyone. The system may also assist in reducing the financial overheads of security auditing and maintenance.

**Challenges and Future work**

The paper presents the discussion and development of an expert system architecture that can help in increasing protection of data within a cloud platform. This section describes the challenges and future work that needs to be done for the completion of proposed expert system.

In pursuing this development, there are many formidable challenges to overcome. First, we need to consider how to ensure that the inputs, both manual and automated, are sufficient to adequately assess a cloud infrastructure’s security. Second, exploring how to check the validity and reliability of the data provided by penetrating testing tools. Furthermore, we need to consider how to determine the root cause of threat. Finally, we must also consider how to minimize false negatives in threat identification without adversely affecting accuracy. Solutions should be proposed while considering the context of current cloud. In addition, we consider how to adjust the functionality of this system to be more objective rather than subjective. I.e. provide solutions which are more reliable.

Future work also includes the development of intelligent forward chaining inference algorithm that can determine threats from attribute/value pairs and solutions from the identified threats. The main challenge is to compare large amounts of data within complex network of knowledge base and deduce the most feasible results.

**Conclusion**

Cloud systems are complex and involve many components working together. There cannot be one-size-fits-all security solution. Like the cloud system itself, the security should also be modularized and multi-layered. A cloud contains huge amount of archival, sensitive and critical data, which is either at rest, in processing or in motion. The proposed expert system will not only help in identifying the security flaws of OpenStack cloud before attacker exploits them, but also propose the most feasible solutions within its scope of knowledge. By introducing a Heuristic Engine in the expert system, we can significantly increase the efficiency, decision making ability and performance of cloud. This expert system can save large amount of human/financial resources and reputation of provider. Another important feature about this system is the support for non-IT users, who can also understand the state of their cloud.

**References**


