Model-Based Security Assessment on the Design of a Patient-Centric Data Sharing Platform

Matthew Banton, Thais Webber, Agastya Silvina, and Juliana Bowles(✉)

School of Computer Science, University of St Andrews, St Andrews KY16 9SX, UK
{tcwds,jkfb}@st-andrews.ac.uk

Abstract. The architectural design of a healthcare data sharing system must cope with security requirements especially when the system integrates different data sources and patient-centric features. The design choices come with different risks, where vulnerabilities and threats highly depend on how the system components interact and depend on each other to operate as well as how it handles the external connections. This paper focuses on security aspects arising early in the design phase of a patient-centric system. The system presents a blend of emergent technologies such as novel authentication methods, blockchain for access control, and a data lake for patient metadata storage and retrieval based on access rules. We exploit a model-based approach to tackle security assessment using attack-defense trees (ADtrees) formalism and other support diagrams altogether as a way to model and analyse potential attack paths to the system and its countermeasures. The modelling approach helps creating a framework to support the attack vectors analysis and the proposal of appropriate defense mechanisms within the system architecture.

Keywords: Healthcare systems · Patient-centric system · Data sharing · Security assessment · Attack-defense trees

1 Introduction

With data breaches on the rise especially after the Covid-19 pandemic [25], the design of robust healthcare platforms leveraging patient-centric features is crucial to allow vital health data to be securely shared among professionals and organisations without leaking patients' private confidential information to any unauthorised user [6].

The research in this paper was supported by the EU H2020 project SERUMS: Securing Medical Data in Smart Patient-Centric Healthcare Systems (grant code 826278).

J. Bowles et al. (Eds.): DataMod 2021, LNCS 13268, pp. 61–77, 2022.
https://doi.org/10.1007/978-3-031-16011-0_5

The EU project Serums[1] [5,6,13,37] proposes a secure patient-centric architectural model integrating modern technologies for user authentication [4,8], granular access to medical records through personalised access rules stored on a blockchain [1], and a data lake for patients' metadata storage and retrieval [37], among other goals. Both blockchain and data lake in the platform are essential components for the access control over medical records, where the former authorises users' data requests in real-time and on-demand, and the later subsequently retrieves only to authorised users the agreed medical data to be shared, running an underlying fine-grained authorisation scheme [6,7].

Serums platform propose features that allow patients full control over their data, which can be stored in different locations, under different data protection regulations and formats [6,18], however, it also opens opportunities for malicious actors to exploit the vulnerabilities to access the system and exfiltrate confidential data or even compromise patient safety [2,22,36]. Attacks in this sense may vary from threatening patients' privacy and data confidentiality, for example, targeting specifically the system authentication module to gain access to sensitive data through social engineering (e.g., phishing attacks), to sniffing the network to intercept traffic using ample techniques and resources, which could also threaten data integrity and availability [31,36].

Model-based security assessments [27] are a viable and visual way of understanding and mapping most likely threats and vulnerabilities of a system. It increases situational awareness and assists modellers in addressing the set of attack vectors (and pathways) at the same time ensuring security controls in place are addressing potential issues that may arise. Several cyber security databases provide information and knowledge on threats, vulnerabilities, tactics and techniques based on real-world observations. MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) [24] is a framework that contains information on adversary behaviour (tactics and techniques), providing a structured view on attack lifecycles and target platforms, helping analysts to prioritise the threats to organisations and systems. Vulnerabilities databases across the world (e.g., CVSS, CVE) map and provide a common interface for security analysts when addressing such shortcomings [23,28].

Among different formalisms to reason and describe security aspects [27], Attack-Defense trees (ADtrees) [12] are an approach that extends the Attack Trees formalism [38], including not only the actions of an attacker, but also possible counteractions of the defender. Its strength is that it provides a broad visualisation of discovering attack vectors and defenses to system architectural designers and developers [12]. ADtrees are also capable of providing quantitative evaluation when the model reaches a reasonable state of maturity and refinement, e.g., calculating a set of measures like probability, cost and time, through a bottom-up procedure implemented in the ADTool software [15].

Focusing more on software architecture, Mal-activity diagrams (MAD), which are a form of UML activity diagram, could be complementary to the ADtree

[1] For more information on Serums project please refer to https://www.serums-h2020.org.

approach since they allow modelling the progression of malicious activities and actors within the system while including information on how the system could defend against these activities [20]. The strength of MAD is the flexibility to detail the model with internal system processes and components interactions. We can therefore use an ADtree to gather an overview of possible attacks and countermeasures, particularly related to adversaries trying to gain access to sensitive data like credentials information through social engineering, then escalating their attack to make use of system features. Using MAD [19] we can show how these attacks and countermeasures would work in practice.

While ADtrees and MAD have been proposed to work together in the past [19], few papers have discussed how to link them, or the process that should be involved in doing so. In this paper, we apply a process that starts with building an ADTree based upon a well-known cyber-adversary behaviour and taxonomy knowledge database (i.e., MITRE [24]) and conclude with a MAD model based upon a threat identified in an initial security assessment [2]. These models are part of an overall security analysis of Serums, which includes formal verification methods [27] and broad security assessment through models [10].

This paper is structured as follows. Section 2 brings background information and related work on the formal modelling of systems focusing on security aspects, especially using ADtrees and MAD. Section 3 presents a brief description on the Serums system design (Subsect. 3.1) and a description of the Serum security assessment using ADTrees (Subsect. 3.2). Section 4 expands the ADtrees scope, modelling MAD to include a more visual description on the system pathways whilst under attack highlighting architectural aspects and mechanisms to mitigate the risk of these attacks. Section 5 presents considerations on the way that modelling activities can demonstrate aspects of security within Serums system, and how they can evolve to serve as basis to quantitative evaluations in future.

2 Background and Related Work

Cyber-security assessment is essential in healthcare systems to keep confidential data safe, especially with the increase in threats during the recent pandemic and the financial impact on the health sector [3,21,25]. Moreover, patients have legal rights in regard to the way healthcare organisations and systems store, access, process, transmit and share their private information [18]. Thus, when designing patient-centric systems for data sharing it is important not only to include security practices and comply with regulations related to data protection such as GDPR, but also conduct different assessments on the system vulnerabilities, threats and security controls enabled by the chosen technologies [31,35].

Being able to combine the above-mentioned requirements (that the system be dependable and secure [27] whilst providing patients full lawful access to their data) requires the data sharing system to be evaluated from both a theoretical and practical level. From the theoretical viewpoint, the system must be able to mitigate common security vulnerabilities and threats. This can be analysed with different security assessment methodologies, using frameworks and widely

recognised security knowledge databases [23,24] but also with diverse modelling practices for security inspections [16,19,27,34]. Typically, multiple complementary modelling approaches together can provide a richer, more comprehensive cyber security assessment on the system under study [26,27].

Several studies using mostly tree-based format [19,26,30,38] have been proposed to model and inspect security aspects in systems, e.g., identifying security threats such as attempts to gain unauthorised access, unsafe system pathways leading to confidential data, among other related security risk analysis [35]. Fault Tree Analysis (FTA) [27] is a traditional method of evaluating the reliability of safety critical systems. The advantages of FTA include being able to identify potential failures deductively, creating a graphical aid for system analysis, and being able to highlight important elements of a system related to the failure. Research on applied FTA for enhanced design of security-critical systems include requirements identification and analysis for an intrusion detection system (IDS) [11] and more recently security failure analysis of smart homes [39].

However, fault tree analysis is both complex, and not suited to security specific situations, i.e., understanding an attackers specific capabilities, or what the life cycle of an attack might include. Further, Attack Trees (AT) [33] introduce a methodical way of describing systems based on the attacks they may encounter. AT provide a method to formally reason about the security of a system, and to capture and reuse the security expertise within the system. They focus more on the mapping of security breaches as system failures, thus modelling possible attacks against the system [26]. They are particularly useful for plotting the progression of possible threats and how to deal with them. The AT root node is the attacker's goal (e.g., information disclosure), and each leaf node is a potential subgoal (or step) the attacker can exploit to reach the root, which means the attack is complete. In the literature, recent applications of AT formalism to assess healthcare systems security include the analysis of IoT devices and their interconnections, just to name a few [14,40].

Attack-Defense Trees (ADtrees) [16] extended AT formalism adding countermeasures to the tree, using green squares for defenses as opposed to the red circles representing attackers. This allows for an intuitive and visual representation of the interaction between possible attacks and defensive measures. The ADTree modelling allows refinement of nodes into sub-goals nodes. These refinements can be either disjunctive (the goal of a node is achieved when at least one of its sub-goals is achieved) or conjuctive (the goal of a node is achieved when all of its sub-goals are achieved; graphically this includes an arc joining sub-goals nodes of a node). Moreover, ADtrees can be also extended to analyse quantitatively attack-defense scenarios and rank possible attacks for given attribute domains using the Attack-Defense Tree Tool (ADTool) [15]. However, this attack-centric view of the system limits the precision defensive strategies can be analysed with, as it does not account for existing defensive strategies within the system. It also does not allow for the visualisation of the evolution of a system's security, since that evolution can only be understood in view of both the attackers and defenders' actions in the tree [16].

Alternatively, from a software design perspective, Misuse Cases [29,34] were applied to encourage the system developer (or software architect) to think like an attacker and represent the requirements of (malicious) users in comprehensible models applying UML-based graphical format. The approach facilitates the communication of system requirements among developers and stakeholders and may lead to the development of a system with decreased cyber security risks. However, such models are not well suited to expand the view on the lifecycle of an attack, thus Guttorm [19,34] proposed Mal-activity diagrams (MAD). MAD provide a way to view the lifecycle of an attack, and the various actions one or more attacks may perform to realise an attack, as well as a way to view these internal interactions. They provide elements to represent the actions performed by users (attackers) to realise an attack within a system, also depicting the existent interactions among components.

Finally, modelling security aspects of systems through different formalisms is an invaluable way to comprehensively study security issues, detecting system's weaknesses and vulnerabilities [27]. We can assume that multiple modelling techniques are helpful in different ways, and that they can be complimentary [9,19,38]. Therefore, it is recommended to use different models and frameworks to assess system security during its design phase [27].

3 Assessment of Attack Scenarios

In this paper we model a high-level cyber security scenario in which we assess the kinds of attacks a particular data sharing system may come under, as well as offer countermeasures. We then expand on this high-level scenario by drilling down into one type of attack (through social engineering) that has been highlighted in a previous research paper [2], with description of four potential threat scenarios concerning both confidentiality and availability aspects of the data sharing system. In this paper, we discuss a threat mostly related to confidentiality, specifically phishing attack scenarios.

3.1 Serums System Design Overview

Serums proof-of-concept (POC) system [13] integrates different software components and technologies to provide a confidential, secure and transparent medical cross-country data sharing platform to users in the healthcare domain. Users are mainly patients and healthcare professionals. Administrators in organisations also make part of the system, contributing to access control over confidential medical records (management of access rules) and creation of new Serums' users. The system should allow organisations to follow laws and regulations (such as GDPR), in a similar manner to how they already do [6].

Figure 1 depicts a high-level overview of the system as a deployment diagram. Serums POC system is a web-based application containing both a front-end and a backend. The front-end consists of both the Serums API, and the SHCS (Smart Health Centre System). The SHCS allows patients users to securely login in

the platform, retrieve own records that are stored in multiple locations (organised by a data lake) through secure data transfers along a blockchain network. They can also create their individual set of access rules, which either enable or deny selected professionals from accessing their confidential medical records. The Serums API alternatively integrates the SHCS with the various technologies being used by Serums in the backend (i.e., the blockchain, data lake and authentication module).

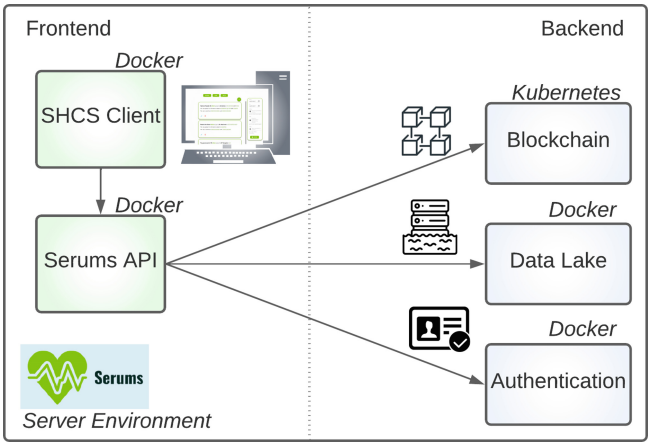

Fig. 1. Serums POC system deployment diagram.

Healthcare professionals, once authenticated in the system (within the SHCS), can search for patients, and retrieve medical records according to their set of access rules provided by patients and administrators. A Serums ID and successful authentication are important requirements to start performing activities in the system. Professionals can also create requests for medical data access to patients, especially to refer to historical data on this patient (i.e., records stored in different locations that are not originally granted access) or current/new collected medical data (e.g., health tracking devices, other health monitoring systems). Patients can promptly accept (or reject) these access requests in the system once they are successfully authenticated in the system.

Internally, the backend is composed of loosely coupled software components in order to fulfil the above-mentioned requirements and provide enhanced security in transactions performed within the system [37]. The authentication module [8] contains novel methods and pictorial passwords, along with traditional textual passwords and other security techniques, to ensure different levels of security checks in this first layer of system access [4]. After it releases a secure access token, functionalities related to medical records retrieval invoke the Serums data lake to process the requests and build a Smart Patient Health Record (SPHR) [5] to the user. SPHR contains authorised health metadata from different sources

(locations), provided the user has conflict-free access rules in place. These rules establish which professional (considering location and authorisation) can have access to what (granular medical data selected by the patient) and when (validity of the rule) [1]. The blockchain component stores and manages the patients' access rules, providing secure authorisation process for data retrievals requested by users (patients and professionals) based on stored access rules [7] with audit trail. Serums POC system aims to demonstrate the interoperability among technologies but also evaluate the public trust and security perception on the proposed solution for healthcare provision.

3.2 Serum Security Assessment Using ADtrees

Serums POC system design presents modern technologies integrated to mitigate several modern cyber threats. During its design, we choose ADtrees [12] as a threat modelling methodology to visualise and analyse potential attacks to the Serums system. ADtrees provide useful construct set capabilities to model the exploit of attack vectors and to propose countermeasures considering top level components of the system architecture.

Our initial assessment published in [2] has provided insights into the possible attacks to Serums system related to malicious users gaining access to confidential medical records. The scenario we are dealing with involves applying phishing technique, which is a known Initial Access tactic employed by adversaries aiming to get into the system [24].

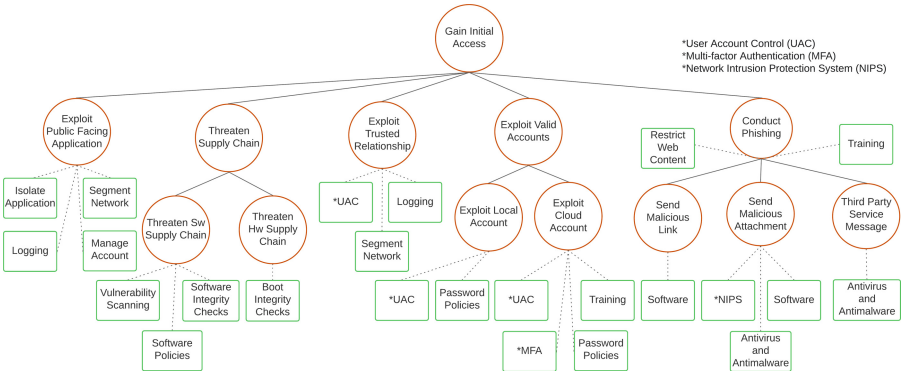

Fig. 2. ADtree for initial access based on MITRE ATT&CK.

The MITRE ATT&CK framework [24] presents a comprehensive list of attack vectors and threat actors in such a well-structured way that enables to integrate the knowledge on unique tactics, techniques, and procedures (TTPs) into the ADtrees. Figure 2 shows an ADTree interpretation of the Initial Access tactic unfolding its techniques and countermeasures.

We have identified possible attack vectors and established which malicious actions (i.e., red circles in the ADtree are adversaries' goals and respective sub-goals) could be performed by their means like exploitation of a public facing application, supply chain, trusted relationships, valid accounts, or Spearphishing. The sub-goals nodes in the ADtree are disjuctive, which means at least one of them, when reached, make the goal achieved. Potential countermeasures (i.e., green squares), especially for phishing, include 'Restricting web-based content', 'Training', 'Software', 'Anti-malware' and 'NIPS' (i.e., Network Intrusion Protection System) to mitigate different types of phishing attempts. It can be seen that, for example, "Threaten Supply Chain" is refined into two potential actions (sub-goals), either hardware (e.g., ensure hardware is distributed with built in vulnerabilities the attacker can access) or software (e.g., create a software update for software used in the system that gives the attacker access to the systems it is installed on). These actions have the potential mitigation applying 'Vulnerability Scanning', 'Software Policies', and 'Integrity Checks'. 'Vulnerability Scanning' helps to ensure that software updates are free from known vulnerabilities, even if distributed by a trusted source, whereas 'Software Policies' can help limit the impact of a successful attack through methods such as 'Sandboxing'.

Meanwhile, integrity checks can help ensure that the software (or hardware) being installed does come from a legitimate source and has been approved by the manufacturer. Primarily we are interested in the phishing technique, which has malicious link, attachment or via third party messages as vectors. Links can be mitigated through software disabling links, and through restricting access to potentially malicious sites as well as training users. Attachments are similar but can also be mitigated through the use of anti-virus software or intrusion detection systems. Third party messaging services (e.g., WhatsApp) are also a risk, and can be mitigated by anti-malware software. Once the malicious actor has gained access to the account, the next step would be to gain persistence through the creation of a 'Healthcare Professional' account, or an 'Administrator' account. Persistence like this could only be gained through compromising an 'Administrator' account, since healthcare professionals and patients do not have the ability to create new other accounts on their behalf, nor it is possible to elevate a patient or professional account to perform administrator role in the system. It is likely that some discovery techniques (e.g., Account Discovery, Password Policy Discovery, Account Control Policies) would also be used at this point by adversaries according to MITRE [24].

The persistence ADTree is shown in Fig. 3. In this case we are concerned with 'Account Creation', although 'Account Manipulation' would also be possible. We can identify promising mitigation being Multi-factor Authentication ('MFA') and 'Account Management'. It would make sense to employ both countermeasures, potentially having MFA second check when a new professional account is created (as this should only happen when a new medical professional starts working at an organisation), as well as having automated tools to track new employees being created, and ways to link them to external data to ensure the employee is real (e.g., using an employee ID). Once the actor has access to

a medical professional account, they would need to employ more Spearphishing attempts, this time defined under Lateral movement (Fig. 4), and internal Spearphishing specifically. This would take the form of asking patients to give the malicious actors account permission to access their medical records.

Internal Spearphishing is difficult to mitigate, since it is based on abuse of system features. As such, training is listed as the main mitigation. It would be important to ensure that patients are informed of who may be expected to access their data and are informed of any requests ahead of time so that unsolicited access requests are unusual and could raise a patients suspicions. Financial institutions often advise their customers on possible fraud and threats by ensuring their correspondence contains phrases such as "We will never call and ask you to provide your bank details."

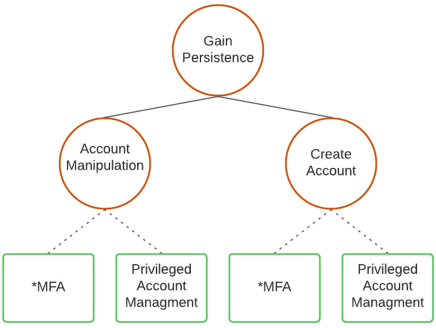

Fig. 3. ADtree for credential access.

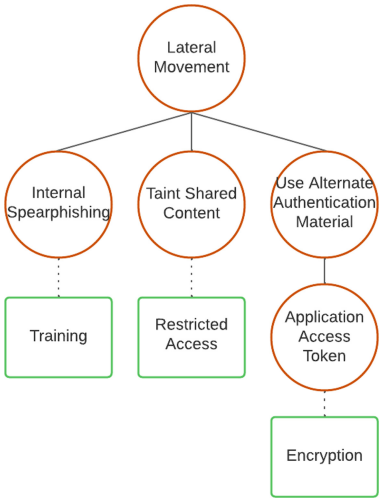

Fig. 4. ADtree for lateral movement.

However, with a healthcare organisation and systems having potentially thousands of users, it is likely that some patients would allow their data to be viewed. Detection is therefore the next best mitigation approach, requiring both network and application logging. In Serums case, the logs could be stored on the blockchain, ensuring that they are not deleted or edited, and that the account making the requests can be identified and any rules created can be undone quickly, or the request removed if the patient has not replied. It is likely the logged information would also feed into a Security Information and Event Management system (or SIEM).

Once patients have given access to the malicious medical account, the actor would need to collect and then exfiltrate the data. The malicious actor would assume the medical account will be identified eventually, and the data will need

to be sent to a location the actor can access it independently of the Serums. This would likely take the form of automated collection, for instance, a script that automatically attempt to access records of any patients a request has been sent to, and if successful, save the data to an external file by reading the archive files (WAR) xml. Again, this activity is difficult to mitigate, since the data is supposed to be accessible by anyone that has been given permission. However, information requests are logged on the blockchain (even if successful or not) and would likely be a part of a SIEM as security measure.

When aggregating these trees together, we obtain the ADtree in Fig. 5. The aggregation is performed by connecting the afore-mentioned trees in the order of the attack, with the goal ('Accessing patient data') being at the top. Requesting patient data (i.e., exfiltrating) is next, followed by the Spearphishing attempt to receive permission to access the data ('D1' and 'E1'). Persistence is gained in 'F1' and 'G1', while the initial phishing attempt is shown in nodes 'H1', 'I1' and 'J1'. We can observe the top of the tree mitigation primarily consist of 'Logging', whereas further down we identify more mitigation options, as the attacks use fewer internal tools. For the ADtree we have discussed, we follow 'Create a Healthcare Professional Account', 'Access to Admin Account', 'Obtain Credentials', 'Obtain Username/Password', 'Social Engineering'. 'Obtain Authorised Device' is a secondary protection, connected to 'MFA', showing 'MFA' is implemented in the system.

An ADtree as an abstract model can represent attack goals and paths in different levels of detail, depending on the security properties being analysed (confidentiality, integrity, and availability), the knowledge on the system design and on the information flow within the system, as well as on the experience to perform threat modeling. It is out of the scope of this paper to analyse the quality of the models we proposed in contrast with security properties, however, we supported our models using a standard framework and documentation for security analysis in such a way that they provide paths semantically meaningful and refinements can be added in further modeling iterations. In future, we plan to quantitatively assess these attack-defense scenarios for Serums, particularly in terms of time, detectability, and impact, seeking to determine a satisfiability attribute (the probability of the scenario succeeding) [15,17].

4 UML-Based Mal-Activity Diagrams

We have expanded the ADTrees into a process, detailing where and how the mitigation's can play a role in diminishing the attack. As stated in Sect. 1 and Sect. 2, Mal-activity diagrams (MAD) are an intuitive way to visualise this process based on UML activity diagram principles [32]. They allow us to observe what vulnerabilities occur in the functional system with coded defenses. The ADTrees in Sect. 3 set the outline for the MAD, with each red circle being an action the attacker performs, and each green square being a potential countermeasure. All that remains is to build the process and insert what the reactions of the system would be, assuming a successful attack at each stage.

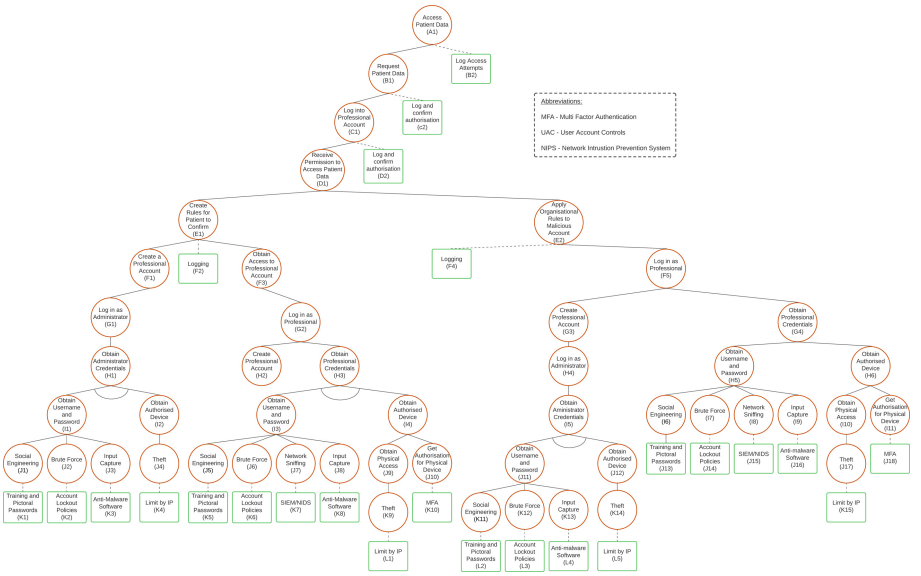

Fig. 5. Integrated ADtree based on MITRE ATT&CK for the 'access of patient data' threat scenario. Nodes containing arcs represent conjunctive sub-goals nodes to achieve a goal node in the tree.

4.1 Mal-Activity Modelling Process

The ADtree provides a broader overview of possible vectors but is not tied to specific attacks (refer to Sect. 2). Thus, extending it to a MAD, we approach a specific attack followed by its vector. As we are tackling the Scenario 2 identified in [2], the vector being followed is one where the malicious user gains control over an 'Administrator' account and uses it to create a 'Professional' account and assign several patients access rules to it. This means following the branch on the left, starting at node 'I1' in Fig. 5. The red circles proceeding up the tree, create a path from 'I1' (then up to 'H1', 'G1', 'F1', and so on), which will then be interpreted as the malicious actions within the MAD. However, some of the paths derived from the ADtree (Fig. 5) are not actions that would be included in the MAD simply because the MAD is more specifically referring to a single attack vector. Of note is node 'H1' ('Obtain Administrator Credentials'), which would also assume the theft of an authorised device mentioned in the MAD (i.e., this would warrant its own diagram). Without this action being shown, nodes 'G1' and 'H1' are essentially the same node. Additionally, not all mitigation strategies are appropriate for the system, and the attacker behaviour modelled in some of the ADtree branches will not need to be replicated in the MAD. Within the ADtree, the mitigation nodes are generic responses, not suited to a design view of the system itself. Within the MAD, we can design where and when specific mitigation strategies are implemented and integrated into the system. Figure 6

and Fig. 7 show the attack process, including the various components that must be targeted to successfully perform the attack as described in Sect. 3.

In Fig. 6 we start with an assumption that admin's credentials have been leaked, and this is protected against with training specifically, however as Fig. 2 shows, it is possible to have other defenses, including 'Software' and 'Limited web access'. The attacker (malicious user) needs to log into the system, which has protections of MFA and being location limited. This step is shown more explicitly in Fig. 5, however, it is difficult to stop the process since it is an intended use of the system. The next step is to gain persistence by creating a professional user. From Fig. 3 we added the potential mitigation for this step, which include 'MFA' and 'Privileged Account Management'. The mitigation included in Fig. 6 shows 'Account Management' with secondary proof being required before a new privileged account can be created. Credentials are also checked (via checking the authentication token). MFA is not included here, since it was included in the previous step (logging into the system). It should also be noted that all steps so far have been logged on the immutable blockchain.

The next stage begins with a choice for the attacker. They can either apply organisational rules to the Doctor using the Admin account, or they can start a new Spearphishing campaign, on this opportunity using Serums' own tools to get users to give the attacker access to their data. In both cases the new rules are logged. In Sect. 3 we discuss the route of requesting access from users using Serums own tools. The potential protection includes ensuring that patients are aware of when they should and should not be expecting requests for access to their data (which is included in Fig. 7). This is connected to Fig. 4, however

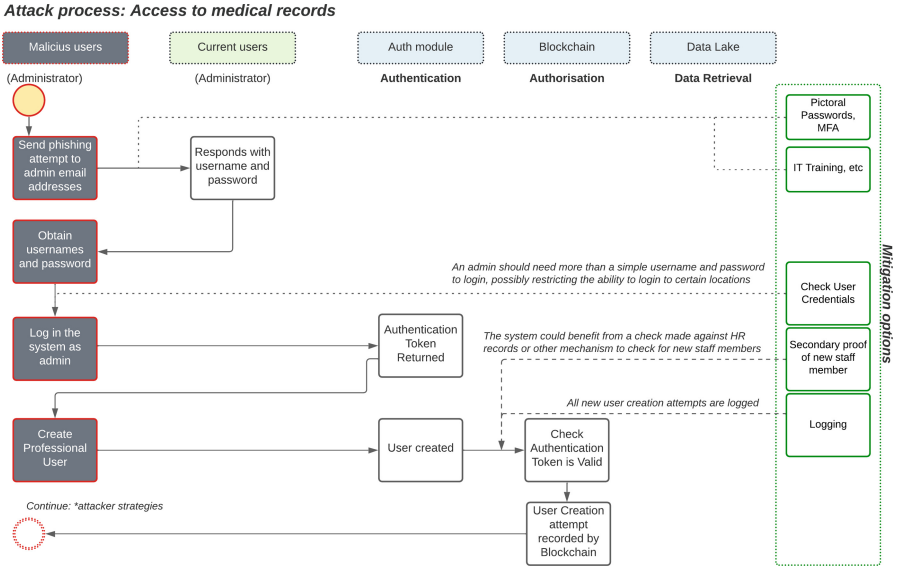

Fig. 6. Mal-activity diagram based on the ADtree on Fig. 5.

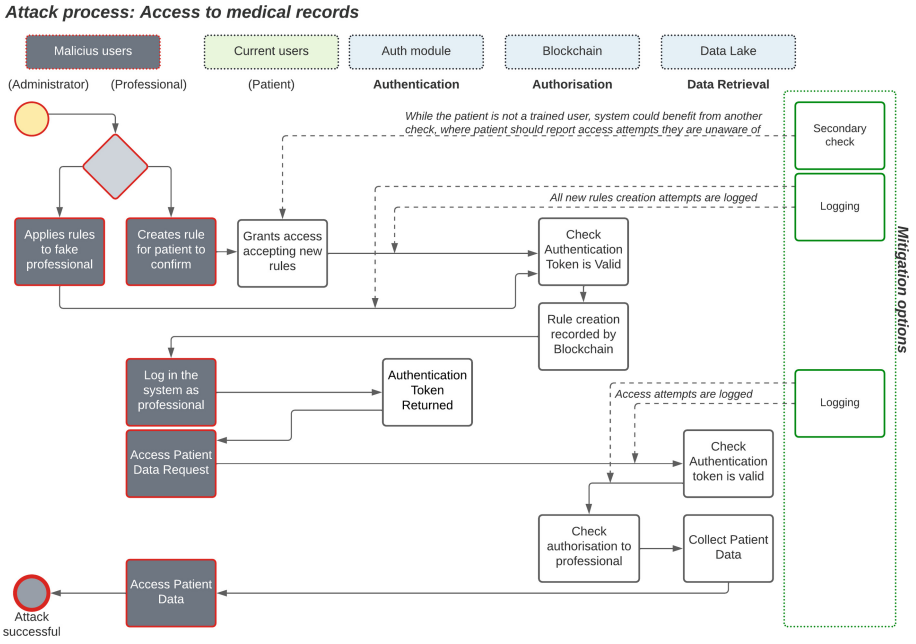

Fig. 7. Mal-activity diagram based on the ADtree on Fig. 5 (cont.).

as mentioned in Sect. 3.2, this would be difficult to mitigate since it is hard to differentiate from normal system use and the steps of an attack, unless a mass campaign was launched that could be detected by a SIEM. Another action difficult to detect would be data collation. The attacker would only need to log into Serums as the professional user and access the data, potentially using external browser-based tools to automatically collate it.

We can now observe the proposed countermeasures, and how they fit into the system overall. For every action the attacker performs, there is a log or a secondary check to attempt to ensure the action is legitimate. We can also see the process maps well from the ADtree (Fig. 5). For example, on creating a new user, the countermeasure listed is to have a secondary check to ensure new users are legitimate. In Fig. 6 we reinforce this with the countermeasure 'Secondary proof of new user'. From the combination of these models, we can observe broadly (using the ADtree) that the attack processes should be discoverable, and from the MAD we can review how these checks would work in practice.

5 Conclusion

Model-based approaches such as ADtrees and other rich diagrams like MAD together can be a way to model and analyse potential attack paths and its countermeasures within systems. They surely expand the knowledge on vulnerabilities

and help demonstrate how platforms could reduce security risks. ADtrees and MAD can support in expanding initial security assessments providing a broad visualisation of the system pathways, its vulnerabilities and defenses, and can be used as a base to design software architectures that are more resilient to attack. Where the strength of ADtrees is to provide a broad visualisation of attack vectors and defenses, the descriptive potential of a MAD is to provide a close view of single components and system flow when under attack. Related work already argued that these are complimentary models that provide holistic view of the system design leveraging security requirements, enabling the early discovery of attack vectors, intrinsic detail of system processes, and responses against attacks [20].

Within this paper we have practised the process on how to construct the ADtree for a system, based upon a security knowledge database like MITRE [24] to produce a sound model, and then used this ADtree as input to model a MAD. With a detailed MAD we could then further refine and build other useful diagrams to describe the system architecture, or even to help developers visualise the activities that need to be completed to fulfil the system security requirements. We have performed this process considering an identified attack for the Serums system, and this has aided identifying its security measures (focus on users training, the need for extensive log analysis, probably as part of a SIEM). In future we intend to cover other aspects of security and apply this modelling process to other attack types. At present we have only shown one attack type and vector, and it would be beneficial to show the process has broad applicability. Finally, we would like to include a quantitative evaluation of the ADtrees especially to prioritise the required MAD. Quantitative evaluations of ADtrees have been discussed by [17], and we believe that such an evaluation would give further inputs to build secure systems.

References

1. Banton, M., Bowles, J., Silvina, A., Webber, T.: Conflict-free access rules for sharing smart patient health records. In: Proceedings of the 5th International Joint Conference on Rules and Reasoning (RuleML+RR 2021). LNCS, vol. 12851, pp. 1–15. Springer (2021). https://doi.org/10.1007/978-3-030-91167-6
2. Banton, M., Bowles, J., Silvina, A., Webber, T.: On the benefits and security risks of a user-centric data sharing platform for healthcare provision. In: UMAP 2021 Adjunct: Publication of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 351–356 (2021). https://doi.org/10.1145/3450614.3464473
3. BBC, O.: Cyber attack 'most significant on Irish state' (2021). https://www.bbc.co.uk/news/world-europe-57111615. Accessed 16 Feb 2022
4. Belk, M., Fidas, C., Pitsillides, A.: FlexPass: symbiosis of seamless user authentication schemes in IoT. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA (2019). http://orcid.org/10.1145/3290607.3312951
5. Bowles, J., Mendoza-Santana, J., Vermeulen, A.F., Webber, T., Blackledge, E.: Integrating healthcare data for enhanced citizen-centred care and analytics. Stud. Health Technol. Inform. **275**, 17–21 (2020). https://doi.org/10.3233/SHTI200686

6. Bowles, J., Mendoza-Santana, J., Webber, T.: Interacting with next-generation smart patient-centric healthcare systems. In: UMAP 2020 Adjunct: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 192–193, July 2020. https://doi.org/10.1145/3386392.3399561
7. Bowles, J., Webber, T., Blackledge, E., Vermeulen, A.: A blockchain-based healthcare platform for secure personalised data sharing. Stud. Health Technol. Inform. Public Health Informat. **281**, 208–212 (2021). https://doi.org/10.3233/SHTI210150
8. Constantinides, A., Belk, M., Fidas, C., Pitsillides, A.: Design and development of the Serums patient-centric user authentication system. In: UMAP 2020 Adjunct: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 201–203, July 2020. https://doi.org/10.1145/3386392.3399564
9. Fraile, M., Ford, M., Gadyatskaya, O., Kumar, R., Stoelinga, M., Trujillo-Rasua, R.: Using attack-defense trees to analyze threats and countermeasures in an ATM: a case study. In: IFIP Working Conference on The Practice of Enterprise Modeling, pp. 326–334. Springer (2016). https://doi.org/10.1007/978-3-319-48393-1
10. Given-Wilson, T., Legay, A.: Formalising fault injection and countermeasures. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES 2020. ACM, New York, NY, USA (2020). https://doi.org/10.1145/3407023.3407049
11. Helmer, G., Wong, J., Slagell, M., Honavar, V., Miller, L., Lutz, R.: A software fault tree approach to requirements analysis of an intrusion detection system. Requirements Eng. **7**(4), 207–220 (2002). https://doi.org/10.1007/s007660200016
12. Hermanns, H., Krämer, J., Krčál, J., Stoelinga, M.: The value of attack-defence diagrams. In: International Conference on Principles of Security and Trust, pp. 163–185. Springer (2016). https://doi.org/10.1007/978-3-662-49635-0
13. Janjic, V., et al.: The serums tool-chain: ensuring security and privacy of medical data in smart patient-centric healthcare systems. In: 2019 IEEE International Conference on Big Data, pp. 2726–2735. IEEE, Los Angeles, CA, USA, December 2019. https://doi.org/10.1109/BigData47090.2019.9005600
14. Kammüller, F.: Combining secure system design with risk assessment for IoT healthcare systems. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 961–966. IEEE (2019). https://doi.org/10.1109/PERCOMW.2019.8730776
15. Kordy, B., Kordy, P., Mauw, S., Schweitzer, P.: ADTool: security analysis with attack-defense trees. In: International Conference on Quantitative Evaluation of Systems (QEST), pp. 173–176. Springer (2013). https://doi.org/10.1007/978-3-642-40196-1
16. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack-defense trees. In: Degano, P., Etalle, S., Guttman, J. (eds.) Formal Aspects of Security and Trust. FAST 2010. LNCS, vol. 6561, pp. 80–95. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19751-2
17. Kordy, B., Mauw, S., Schweitzer, P.: Quantitative questions on attack-defense trees. In: International Conference on Information Security and Cryptology, pp. 49–64. Springer (2012). https://doi.org/10.1007/978-3-642-37682-5
18. Larrucea, X., Moffie, M., Asaf, S., Santamaria, I.: Towards a GDPR compliant way to secure European cross border healthcare industry 4.0. Comput. Stand. Interf. **69**, 103408 (2020). https://doi.org/10.1016/j.csi.2019.103408

19. Löhner, B.: Attack-defense-trees and other security modeling tools. In: Niedermayer, H. (ed.) Network Architectures and Services, Seminar Future Internet, pp. 97–103 (2018). https://doi.org/10.2313/NET-2018-11-1
20. Mai, P.X., Goknil, A., Shar, L.K., Pastore, F., Briand, L.C., Shaame, S.: Modeling security and privacy requirements: a use case-driven approach. Inf. Softw. Technol. **100**, 165–182 (2018). https://doi.org/10.1016/j.infsof.2018.04.007
21. McKeon, J.: KY Hospital Systems Still Down 1 Week After Cybersecurity Incident, Health IT Security, xtelligent Healthcare Media (2022). https://www.healthitsecurity.com/news/ky-hospital-systems-still-down-1-week-after-cybersecurity-incident. Accessed 16 Feb 2022
22. Meingast, M., Roosta, T., Sastry, S.: Security and privacy issues with health care information technology. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5453–5458. IEEE (2006). https://doi.org/10.1109/IEMBS.2006.260060
23. MITRE Corporation: Common vulnerability and exposures, https://cve.mitre.org/. Accessed 16 Feb 2022
24. MITRE Corporation: MITRE ATT&CK, https://www.attack.mitre.org/. Accessed 16 Feb 2022
25. Muthuppalaniappan, M., Stevenson, K.: Healthcare cyber-attacks and the COVID-19 pandemic: an urgent threat to global health. Int. J. Qual. Health Care **33**(1), mzaa117 (2021). https://doi.org/10.1093/intqhc/mzaa117
26. Nagaraju, V., Fiondella, L., Wandji, T.: A survey of fault and attack tree modeling and analysis for cyber risk management. In: 2017 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1–6. IEEE (2017). https://doi.org/10.1109/THS.2017.7943455
27. Nicol, D., Sanders, W., Trivedi, K.: Model-based evaluation: from dependability to security. IEEE Trans. Depend. Secure Comput. **1**(1), 48–65 (2004). https://doi.org/10.1109/TDSC.2004.11
28. NIST Information Technology Laboratory: National vulnerability database (nvd), https://www.nvd.nist.gov/vuln. Accessed 16 Feb 2022
29. Opdahl, A.L., Sindre, G.: Experimental comparison of attack trees and misuse cases for security threat identification. Inf. Softw. Technol. **51**(5), 916–932 (2009). https://doi.org/10.1016/j.infsof.2008.05.013
30. Piètre-Cambacédès, L., Bouissou, M.: Beyond attack trees: dynamic security modeling with Boolean logic driven Markov processes (BDMP). In: 2010 European Dependable Computing Conference, pp. 199–208. IEEE (2010). https://doi.org/10.1109/EDCC.2010.32
31. Priya, R., Sivasankaran, S., Ravisasthiri, P., Sivachandiran, S.: A survey on security attacks in electronic healthcare systems. In: 2017 International Conference on Communication and Signal Processing (ICCSP), pp. 691–694. IEEE (2017). https://doi.org/10.1109/ICCSP.2017.8286448
32. Rumbaugh, J., Jacobson, I., Booch, G.: Unified Modeling Language Reference Manual, The (2nd Edition). Pearson Higher Education (2004)
33. Schneier, B.: Attack trees. Dr Dobb's J.-Softw. Tools. Profess. Programm. **24**(12), 21–31 (1999). https://www.cse.sc.edu/zeng1/csce790-f21/papers/attacktrees.pdf
34. Sindre, G.: Mal-activity diagrams for capturing attacks on business processes. In: Sawyer, P., Paech, B., Heymans, P. (eds.) Requirements Engineering: Foundation for Software Quality, pp. 355–366. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73031-6

35. Souppaya, M., Scarfone, K.: Guide to data-centric system threat modeling. Technical report. Draft NIST Special Publication 800–154, National Institute of Standards and Technology (2016). https://www.csrc.nist.gov/publications/detail/sp/800-154/draft
36. Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M.A., Rashid, A.: Data exfiltration: a review of external attack vectors and countermeasures. J. Netw. Comput. Appl. **101**, 18–54 (2018). https://doi.org/10.1016/j.jnca.2017.10.016
37. Webber, T., Santana, J.M., Vermeulen, A.F., Bowles, J.K.F.: Designing a patient-centric system for secure exchanges of medical data. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12254, pp. 598–614. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58817-5_44
38. Wideł, W., Audinot, M., Fila, B., Pinchinat, S.: Beyond 2014: formal methods for attack tree-based security modeling. ACM Comput. Surv. **52**(4), 1–36 (2019). https://doi.org/10.1145/3331524
39. Wongvises, C., Khurat, A., Fall, D., Kashihara, S.: Fault tree analysis-based risk quantification of smart homes. In: 2017 2nd International Conference on Information Technology (INCIT), pp. 1–6 (2017). https://doi.org/10.1109/INCIT.2017.8257865
40. Xu, J., Venkatasubramanian, K.K., Sfyrla, V.: A methodology for systematic attack trees generation for interoperable medical devices. In: 2016 Annual IEEE Systems Conference (SysCon), pp. 1–7. IEEE (2016). https://doi.org/10.1109/SYSCON.2016.7490632