Protecting Web-based Patient Portal for the Security and Privacy of Electronic Medical Records

Xiaowei Li[1] and Yuan Xue[1]

[1]Department of Electrical Engineering & Computer Science, Vanderbilt University

1 Motivation

Patient portals, such as Myhealthatvanderbilt.com, AdjuvantHealth, Medical Web Experts, are web-based systems that allow both physicians and patients to access and manage patient medical records via the Internet, facilitate clinical workflow and enable data sharing and collaboration. Serving as the front-end to a huge amount of sensitive information (e.g., medical records, billing), the patient portal is an essential link in the chain of ensuring patient data security and privacy. In August 2000, it was reported that over 800 patients' sensitive information was breached through KPOnline, a web-based healthcare portal, due to a small piece of flawed software that is integrated into the system [3].

Development and deployment of a secure web-based patient portal is challenging. First, as a web-based application, the patient portal suffers from all the common security pitfalls, such as weak authentication, cross-site scripting, SQL injection, where traditional defenses, e.g., SSL, firewall, become lame even useless. Nowadays, patient portals are built with an increasing number of web technologies, e.g., Ajax, PHP, CSS, leading to a more complicated landscape. Second, the patient portal has to implement and enforce complex security policies to restrict the access of sensitive information and actions. Such policies are usually dynamic, related with clinical workflows, thus cannot be explicitly or precisely defined. Moreover, the fact that patient portals are usually connected and integrated with other clinical components, including billing, prescription ordering, third-party services, etc., confounds the problem further, when sensitive information (e.g., secret tokens, diagnosis code) is exchanged among them. It is very likely that certain design or implementation flaws are introduced that result in the leakage of sensitive information. The attacker may obtain valuable information via side-channels [1], exploiting subtle logic flaws within their interactions, etc.

2 Case Study: OpenEMR

We select OpenEMR, one of the most popular open source electronic health records and medical practice management applications, to study the vulnerabilities that a patient portal may contain and the consequences if they are exploited. OpenEMR is ONC Complete Ambulatory EHR Certified, developed and supported by a community of volunteers and at least 11 companies. OpenEMR includes a set of customizable components, such as patient demographics, scheduling, medical billing, prescriptions, etc., and supports fine-grained user access control based on the php-GACL module.

However, OpenEMR has been reported to contain a number of security vulnerabilities [6]. Most of them are input validation vulnerabilities, such as cross-site scripting, SQL injection. More disappointingly, Austin et. al. from NCSU found that OpenEMR v3.2 contains a serious authorization flaw [7]. An attacker can log into the application as a regular user (e.g., a receptionist), then creates a new administrator user by directly pointing to the *user_add* page. The root cause is that developers hide the link to the *user_add* page from regular users, but fail to check the user privilege/role when the *user_add* page is accessed and executed. The *patient_id* parameter within a web request is another vulnerable spot that attracts parameter manipulation attack. If the application fails to check the semantic of the parameter with the context, the attacker can retrieve other patients' information that he is not authorized for. Different from input validation vulnerabilities, which can be captured via a general specification based on information flow model, the above flaws are specific to the intended business logic of a particular patient portal and much difficult to identify. Therefore, even for a certified patient portal, it is still possible that sensitive clinical information can be leaked. Since input validation attacks can be effectively thwarted by employing techniques already proposed in the web security area, in this research we focus on securing patient portals from business logic flaws.

3 Proposed Approach

Since patient portals are usually complex and tightly coupled with other components of clinical information systems, trying to identify and patch the vulnerabilities within can be extremely difficult and error-prone. To protect the web-based patient portal and ensure the security and privacy of electronic medical records, we propose to build a defense architecture with two security checkpoints: 1) Request Blocker sitting between the users and the patient portal, which detects the web-request-level attacks and prevents sensitive information to be revealed to the attackers in web responses and 2) EMR Protector, which isolates the potentially vulnerable patient portal from the EMR database by identifying and blocking SQL-query-level attacks. Both Request Blocker and EMR Protector operate over a central decision engine based on a repository of security specifications. The decision engine has access to the user session information through connectors, which can read local session files (e.g., PHP) or retrieve information from database tables (e.g., JSP), thus is capable of evaluating the requests and queries in a context-aware manner and identifying business logic attacks. We adopt a dynamic analysis method to automatically infer the security specifications from the interactions between the patient portal, web clients, EMR database and other components either during the training phase or normal attack-free executions. The inferred security specifications capture the intended behavior of the web-based patient portal and will be verified and associated with clinical semantics by security experts to suppress false positives. The security specifications can be expressed in two forms: rule-based and evidence-based. Rules represent security policies in a deterministic manner (e.g., access control list), while evidence-based policies are designed for expressing more complex and dynamic scenarios based on certain statistical measures (e.g., CADS based on social network analysis of EMR users [2]). Evidence-based specifications with statistical measures can effectively handle the complex and dynamic policies that are unique to the clinical environment, which are usually not predefined, or in some cases, unable to be defined accurately.

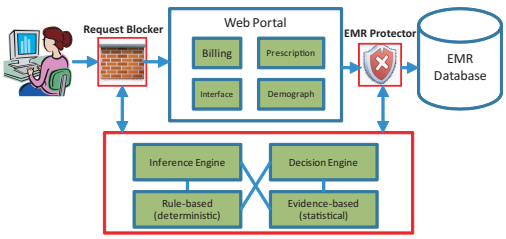

Figure 1: Two-tier Defense Architecture for Web Patient Portal

Our architecture has several benefits. First, security implementation, which is performed by examining the interactions between the patient portal, the EMR database and other third party components, is independent of the specific patient portal, thus can be easily verified and imported for other patient portals. Second, our automatic specification inference mechanism scales well for complex business logic within patient portals and can gracefully handle policy dynamics. Moreover, new security mechanisms can be transparently integrated into our architecture to enhance the protection.

Currently, Request Blocker and EMR Protector are built based on our BLOCK [4] and SENTINEL [5] systems respectively, which are able to support the automatic inference of rule-based security specifications and the detection of logic attacks. For example, in the case of OpenEMR, one security invariant that can be inferred is that the user add page can only be accessed when the role of current user is administrator, indicated by session variables. Another security invariant is that when the SQL query is issued to retrieve the user's patient record, the patient id must be equal to the current user id. In this way, we can detect the attacks that try to bypass authorization checks or manipulate the parameters. In the future, we will enhance the inference engine with evidence-based rules based on statistical measures to handle complex and dynamic clinical information access policies.

Acknowledgements

This work was supported by NSF TRUST (The Team for Research in Ubiquitous Secure Technology) Science and Technology Center (CCF-0424422).

References

[1] CHEN, S., WANG, R., WANG, X., AND ZHANG, K. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Oakland 10* (2010), pp. 191–206.

[2] CHEN, Y., AND MALIN, B. Detection of anomalous insiders in collaborative environments via relational analysis of access logs. In *CODASPY'11* (2011), pp. 63–74.

[3] COLLMANN, J., AND COOPER, T. Breaching the security of the kaiser permanente internet patient portal: the organizational foundations of information security. *Journal of American Medical Informatics 14* (2007), 239–243.

[4] LI, X., AND XUE, Y. BLOCK: a black-box approach for detection of state violation attacks towards web applications. In *ACSAC '11* (2011), pp. 247–256.

[5] LI, X., YAN, W., AND XUE, Y. SENTINEL: securing database from logic flaws in web applications. In *CODASPY '12* (2012), pp. 25–36.

[6] OPENEMR MULTIPLE VULNERABILITIES. http://www.exploitsdownload.com/search?q=emr.

[7] SMITH, B., AUSTIN, A., BROWN, M., KING, J. T., LANKFORD, J., MENEELY, A., AND WILLIAMS, L. Challenges for protecting the privacy of health information: required certification can leave common vulnerabilities undetected. In *SPIMACS '10* (2010), pp. 1–12.