EECE 412 – Security Analysis of EASports.com

Maxime Perreault, David Rosberg, Peter Vautour, David Wang
maxime.perreault@gmail.com -- drosberg@gmail.com -- peterv@ece.ubc.ca -- 18davidwang@gmail.com

***Abstract*—This report is a security analysis of EASports commercial website which attempts to identify security vulnerabilities. This analysis tests for many of the common vulnerabilities found on websites, and includes additional tests of their content management system. Most of the tests did not reveal security problems, with the exception of the content management system where some oversights during setup provided opportunities for many types of attacks. The details of all of these tests are presented. A discussion of EASports security risks, the web sites adherence to common security principles, and its privacy certification follows. This analysis was responsible for the identification and correction of a major exploit.**

***Index Terms*—**
CMS – Content Management System

I. Introduction

The intent of a security analysis is to identify potential vulnerabilities which can potentially be exploited in order to cause damage or steal information. The final goal of an analysis is to not only find these vulnerabilities, but also to correct them.

The internet is full of websites that were hastily created with little or no thought put into security. For those individuals looking to steal personal information, spread spam email, or infect computers with malware, the internet is rich with opportunities and there is a lot of illegal money to be made. A website security analysis is one way to limit these opportunities.

For users of insecure websites, they risk identity theft, which can lead to further financial losses if credit card or banking information is collected directly or indirectly. Users also risk having their computers infected with some form of malware. For corporations, who use unsecure websites to promote their name, products, or conduct online sales, they risk their company's reputation, their customer's goodwill and trust, and sales. All of these leading to a loss in profits, the life blood of most companies.

The website chosen for this security analysis is EASports.com. EASports.com is a large enterprise website with potential risks to both its users and itself. The website boasts over 14 million users, making the risk proportionately large. EASports.com is the marketing website for the collection of games produced by EA Sports. Many financial transactions are made as well. As such, any downtime results in loss of revenue. Furthermore, it is also the community site for their games and displays some properties common to social networking sites. This also increases the risk of attacks like XSS since there are numerous user inputs.

Those seeking to exploit website vulnerabilities have a common set of techniques at their disposal. So, in doing a security analysis, this is a likely place to start. What follows in section II of this report is a discussion of these common attacks, and even though they failed to expose any security vulnerabilities, the analysis procedure would have been incomplete had they been neglected. EASports.com uses a content management system (CMS) to manage their online website; this was another point of interest during the analysis and, as a result, a major security vulnerability was revealed. That vulnerability is discussed at the end of section II.

Section III discusses the potential security risks specific to EASports.com and their website users. There are a set of 10 security principles that should be followed in order to make a secure website, section IV discusses which particular principles were violated by EA.

EASports.com has been TRUSTe certified as a secure website and section V discusses what this certification is and its implications. EASports.com was notified of its security problems and they have taken measures to correct the problems. Section VI discusses what measure they have taken. Finally, there are some concluding remarks in section VII.

II. Steps taken to identify security vulnerabilities

A. SQL Injection

This attack relies on the fact that user input is inserted into a preformed SQL statement. Ordinarily the user input is interpreted as a string, but a malicious user can write a statement that will be interpreted as more than just data, which can potentially cause grave harm. This can enable the malicious user to bypass authentication, read private information, or change and delete information on the database. This attack is easily solved by filtering out potentially offending characters from the input, ensuring that it is nothing more than a string and cannot be interpreted as a SQL statement. This fix is only as strong as its weakest link; every single user input field on the entire site must have this filtering.

When analyzing EASports.com for SQL Injection vulnerabilities, the site would not respond to any overt SQL statements put into these fields, indicating that some form of validation occurred on the server side. The error messages provided by the site after unsuccessful SQL attacks (if any) were generic and did not reveal any technical details.

B. Cross-site Scripting

Sometimes, it is possible for a user to inject script, such as JavaScript, into a website to enable an otherwise legitimate and safe page to run something insecure. This occurs when a web designer presumes that the user will supply a text input, but does not consider the possibility that the text may be interpreted as a script. This could potentially put other users at risk or cause some form of damage to the site. Cross-site scripting can be used for many purposes, such as stealing the session ID's of other users. No exploits of this type were found; EASports.com would not run or display scripts provided as inputs. As with SQL injections, there was some form of validation occurring on the server. Other variations of the input were attempted to bypass the validation without success.

C. AJAX

Using AJAX in web applications enables web applications to do more for the user. Due to a combination of client-side and server-side processing, AJAX applications can be more responsive and flexible. However, the increased complexity may also introduce security vulnerabilities. Because data transfer isn't limited to GET and POST commands of regular HTML, there are more potential holes for attackers to exploit. An analysis of several sessions where AJAX requests and responses occurred yielded no results.

D. Session Management

After authentication, a server will issue a “session ID” to the user. This session ID is intended to be unique and has a limited lifespan before it expires. If a malicious user is able to acquire someone's session ID, they will be able to impersonate that user until the session is ended or times out. Sometimes session ID's are particularly weak or last particularly long, giving a reasonable opportunity for an attacker to acquire a valid one without having to actively steal it from the victim. The session ID's generated by EASports.com were both sufficiently long and random; effectively minimizing the chances of randomly guessing a valid ID.

E. Insecure Configuration

A web server, by default, is exposed to the Internet and has inherent security risks. Security firmware needs to be installed and regularly updated, and all aspects of the site need to be configured properly. It only takes one poorly configured security setting to allow access to a malicious user. Even something as innocuous as an overly informative error message may direct an attacker's attention towards a potential weakness.

The EASports website had a single severe vulnerability in its configuration: the content management system (CMS) was exposed and accessible to the public. Only a single password prompt separated the user from being able to directly access the CMS. This vulnerability will be discussed in greater detail in the following section.

F. Insecure Communication

It is possible that sensitive information may be intercepted over the internet. Using a proper security protocol to prevent this from happening is critical. The use of SSL for any sensitive traffic – not just for authentication – is necessary to maintain the integrity and confidentiality of this data. As a website that makes financial transactions, EASports.com routinely handles valuable information such as credit card numbers. By inspecting incoming and outgoing packets, it was confirmed that all sensitive information was encrypted and it would be very difficult for an attacker to discover the contents.

G. Parameter Tampering

HTTP was originally designed to be stateless. Web developers creating a web application have the difficult task of maintaining state. The output produced by a page depends entirely on the user's input, cookies and the state of a database, and nothing else. Any information stored or provided by the client-side of the web application may be subject to manipulation by a malicious user; this can be in the form of cookies, hidden fields, or parameters imbedded in the URL. The attack can be stopped by server-side filtering, as it is ultimately the web application that responds to requests. The program “WebScarab” was used to view and tamper with many of these parameters, but opportunistic manipulation of the behavior of the site wasn’t successful. Important information and permissions were adequately maintained on the server-side.

H. CMS

As a major community and marketing website for a large number of sports games, EASports.com must provide its users with fresh content on a regular basis. In order to meet this goal, a content management system known as Alfresco[1] is used. Alfresco as a web content management system allows EA's nontechnical staff to publish content to the website. This added flexibility comes at a price: it introduces another complex system that must be secured. This was not done properly.

The first sign of an exploit came in discovering how content was retrieved. The URLs of their content suggested that it was dynamically queried. For example, a picture of their logo can be found at the following website: http://cdn.content.easports.com/alfresco/service/eaapi/node/content/avm/easportscom/-1;www;avm_webapps;ROOT;;_assets;en_US;sportsworld_logo.png. A little research revealed that Alfresco is a content management system used by EA to serve content. EA is even listed as a customer on Alfresco's home page. The next stage of this project was gathering information about Alfresco. The open source version was downloaded and analyzed. Important paths (i.e. for admin panels), default accounts and passwords, and interfaces that Alfresco provides were noted. With this information in hand, it became evident that the content was retrieved by making calls to an Alfresco webscript. This was the second sign of a major exploit; it was possible to directly execute web scripts on EA's content management system.

Next, the theory that all requests matching the regular expression" /alfresco/**" were redirected to Alfresco was tested. An admin path (/alfresco/service/index) that was found earlier during the research on Alfresco was attempted. The theory turned out to be correct; all requests matching the regular expression above were sent to the CMS without any validation. However, all admin screens can only be accessed by an administrator's account. The default administrator account (admin/admin) was attempted and was found to be valid. Together with the previous weakness, an attacker could install and execute any web script directly against EA's production Alfresco installation. Moreover, the attacker's webscript can run as the admin user; thus there would be no restrictions on the amount of damage possible.

At this point, several flaws had been revealed that would allow an attacker complete access to EA's content. This included news posts, blog posts, pictures, videos, etc. From here, several attacks were crafted to take full advantage of this exploit. Testing these on EA's production server was out of the question, so all work from this point was done on a local installation of Alfresco. It was possible to remove select pieces or the entire set of content. It was also possible to lock the project so no new updates could be made; together with changing the admin password could serve as a denial-of-service attack. Finally, the last attack attempted was inserting arbitrary strings into content whose type was text or html. Alfresco's API allowed us to perform this last attack. This could enable an attacker to launch attacks on the users of EASports.com. The arbitrary string could easily have been a malicious JavaScript program. From there, a whole host of attacks against users are possible (i.e. session stealing).

III. Analysis of Security Risks

Risk is a combination of the probability that an event will occur and the consequences of its occurrence. It can be represented as the following formula:

*Risk = Assets * Threats * Vulnerabilities* [2]

Assets are the tangible and intangible things one owns that could be lost. Threats are the potential means by which loss many occur. The vulnerabilities are the weaknesses in one's existing security measures that allow threats to be successful. [2][3]. To analyze the security risks at EASports, it is important to first identify its assets. Below are the assets that were discovered:

- Brand and reputation
- Client's trust
- Client's personal information (credit card number)
- Content integrity

Then, some of the threats to those assets were identified:

- Password-stealing
- Video sharing to gain popularity
- Inevitable targeting of video to distribute malicious code
- Denial of service
- Session stealing
- Disclosure of private information

These threats could be caused by the following threat agents:

- Students (research purpose)
- Hackers
- Opposing companies
- Dissatisfied customers

Potential sources of vulnerabilities could also be caused by human factors such as:

- Honesty of employees
- Morality of employees
- Knowledge of security
- Habits of safe computing

EASports' large asset value contributes to a much greater risk. Since threats cannot be controlled, the risk can be minimized by coming up with security measures to protect assets. In the next section, some general security principles for designing a secure system will be discussed.

IV. Security Principles

It is generally accepted that there are a set of 10 security principles that should be followed in order to design a secure system. Four of these principles were violated when EASports.com website was implemented.

1. Fail-Safe Defaults

When designing any complex system, it is important to consider that individuals involved in the installation or operation of that system may not understand the inner workings of that system, and more importantly the repercussions that may follow if that system is not set up properly. When it comes to security issues, steps should be taken in order to ensure that security is not left as a secondary consideration. One important step is to make sure that the systems default settings are secure and to not rely on the installers or operators to choose those settings.

The Content Management System that EASports uses did not follow this philosophy. The system has an insecure admin password that needs to be changed at installation time. Although the use of such a default password is necessary at installation time, the system should have a reminder to change the password if the default password was still in use.

2. Least Common Mechanism

The principle of "least common mechanism" states that one should minimize the amount of mechanism common to more than one user and depended on by all users. EASports failed to do so. It completely exposed its CMS to the Internet merely to serve content to its users. As a side effect, any one could access administrative functions. EA should have limited the scope of access to the CMS by providing a separate mechanism to query content.

3. Defense in Depth

It is important to have security mechanisms in place to protect your assets. However, relying on just one mechanism has not always proven safe. If that one mechanism fails, there will be no other barriers to protect your assets. Having multiple levels of protection, or defense in depth, is a much better option. If one system fails, you are still protected.

A simple manipulation of EASports web address was able to trigger the admin password prompt to come up. Having the default admin password still in use made that one barrier easy to bypass. Rather than relying on a single hurdle, there should have been an additional barrier such as URL filtering to prevent indiscriminate access to the CMS. This would have made it far more difficult to gain access to the inner workings of the system.

4. Question Assumptions

With any security features, there are assumptions made about; the environment the system will work in, who will access the system, how the system will be configured, who will operate the system, etc. Although it is not possible to design systems without making some assumptions, it is important to verify and revisit these assumptions from time to time. The reason is simply that they may not be valid in all circumstances and over time, changes may occur that render these assumptions invalid.

It is very likely that, at installation time, it was assumed that the default admin password would be changed before going to production. But, obviously, no one followed up to verify that assumption, and the password was never changed. The website also underwent a security compliance audit where again, it was probably assumed that strong passwords were in place. The next section discusses more about the compliance audit.

V. Truste certification

According to their website [4], TRUSTe is a leader in internet privacy service and has been providing this service since 1997. It was founded as a non-profit organization with the intent of promoting online commerce by aiding businesses and organizations in self-regulating privacy concerns. Out of that came the TRUSTe Privacy Seal, a logo that can be displayed on a website which certifies that the business or organization has met with a list of standards [5] that TRUSTe considers adequate to protect the privacy of individuals using the site. That list requires that the site must protect against unauthorized access. Websites displaying the logo must undergo regular compliance monitoring to ensure that the site not only initially meets the standards, but continues to uphold them.

The goal, of course, is that users of a certified website can assume that their private information is kept secure. As stated on the TRUSTe website; “Web sites can build trust with their customers and increase sales and registrations” [PV1]. This assumption, based on this security analysis, may not be a very sound one. Although the compliance monitoring did ensure that most of our efforts to uncover security vulnerabilities met with failure, it fell short in ensuring that EASports was acting responsibly when it came to accessing their own content management system. This failure effectively undermined all other efforts to secure the website.

TRUSTe, perhaps, needs to extend its compliance monitoring beyond the purely technical aspects of securing a website, to ensure that the operators of the site are acting responsibly.

VI. Corrective Measures

The details of this exploit were communicated to the appropriate parties in EA. The exploit was fixed within 24 hours of disclosure; this further highlights the severity of this exploit. Later, it was confirmed that the exploit was indeed removed. URL filtering was now being done and only certain queries were now being redirected to the CMS. As such, it was no longer possible to access any administrative interfaces.

VII. Conclusion

This analysis revealed several facets of enterprise web security. Large investments go into fixing popular types of attacks. We couldn't find a single traditional exploit. However, security of the web site is only as strong as its weakest link. In this case, gaining access to the CMS enabled a host of various attacks. Moreover, enterprises have a large budget and can afford spending time and money on security. Most websites don't have this luxury. Despite the efforts in securing EASports.com and the fact that it is TRUSTe certified, we were still able to uncover a major security flaw. This may be due to the dynamic nature of our particular target; frequent changes are required due to new releases and the need to provide users fresh content. In general, this illustrates the complexities and challenges in securing a given web application.

References

[1] Alfresco, Nov 27, 2009. [Online]. Available: http://www.alfresco.com [Accessed: Nov 27, 2009].

[2] R. Anderson, Security Engineering, John Wiley & Sons, New York, 2001.

[3] The Center for Information Systems Security Studies and Research. Nov 27, 2009. [Online]. Available: http://cisr.nps.edu/downloads/nps_cs_05_010.pdf [Accessed: Nov 27, 2009]

[4] Truste home page. Nov 07, 2009. [Online]. Available: http://www.truste.com/ [Accessed: Nov 27, 2009]

[5] Truste, Privacy Program Requirements, Nov 07, 2009. [Online]. Available: http://www.truste.com/privacy_seals_and_services/consumer_privacy/privacy-programs-requirements.html [Accessed: Nov 27, 2009].