Final Project

Submission policy. All parts of the final project are due via websubmit, following the submission checklist below. Your submission MUST include the following information:

1. List of collaborators (on all parts of the project, not just the writeup)
2. List of references used (online material, course nodes, textbooks, wikipedia, etc.)

If any of this information is missing, at least 20% of the points for the assignment will automatically be deducted from your assignment. See also discussion on plagiarism and the collaboration policy on the course syllabus.

1 Introduction

In this project you will choose a website and analyze the client-server communication from the website, in order to understand what information is collected and recorded when users surf the web. Your primary tool will be the developer tools bundled with the Chrome browser.
You will be analyzing and interacting with production systems in this project. Please make sure you read and understand the rules given below before beginning the assignment. Failure to follow the rules will result in an automatic F in this course.
CS Masters students who wish this project to be their CS Masters project should work alone. All other students must work in teams of two or three. The project has three deliverables:

1. Writeup with, choice of group, website, and analysis of the website's responsible disclosure policy. **Due: April 10, 2014 at 11:59PM.**

2. Disclosure, to Professor Goldberg, of any potential vulnerability that you might have stumbled upon as part of your work. This should be done verbally during Prof Goldberg's office hours. **Due: Prof Goldberg's office hours, during the week of April 27, 2015 (or earlier).**

3. Poster session and report. **Due: Friday, May 1, 2015, 9AM.**

Administration. The technical details of this project (how to use the Chrome developer tools, questions about cookies, html, javascript, etc) will be administrated by Ethan Heilman. Issues related to disclosing vulnerabilities, and presentation of the poster or final report, will be administered by Sharon Goldberg.

2 Story

The Intelligence and Security Apparatus (ISA) of the fictional Republic of Lacedaemonia wishes to spy its own citizens. The ISA may use the following methods (in order of increasing costs):

1. Eavesdrop on any unencrypted network traffic, by acting as a Man In The Middle (MITM) on the network.
2. Gain access to the server and learn everything the server knows.
3. Gain access to the client and learn what the client knows.
4. Hack into a certificate authority, steal its keys, and issue a bogus certificate for the target website. Then, use this certificate, along with a MITM attack, to eavesdrop on *encrypted* internet traffic.

As the ISA does not want to get caught (for obvious legal and political reasons). Thus, information that is can learn from passively observing traffic is the most dangerous and should be the primary focus of your analysis.

Additionally, but far less importantly, you should also consider what information the site collects from the user and who the this information is shared with, as the Republic of Lacedaemonia has treaty and legal agreements with many countries that would allow us to subpoena or request these records.

Your job will be to choose an website to analyze. The ultimate goal of this analysis is to determine the information leaked/stored by the web application, the value of this information to the ISA, and finally to catalogue the methods that could be employed to access it.

3 Websites

You should select a website from the following list, or any website that is listed on the following list of websites that support bug bounty programs:

https://bugcrowd.com/list-of-bug-bounty-programs

You may only choose a site that has a responsible disclosure policy and/or a bug-bounty program. If you really want to review a web application that is not on the list below, or on the bugcrowd list, please discuss with this with Professor Goldberg during her office hours; if you do this, you should be ready with information about the sites' responsible disclosure program.

- Airbnb www.airbnb.com, https://www.airbnb.com/help/policies/responsible_disclosure#responsible_disclosure_policy
- ActiveCampaign www.activecampaign.com, http://www.activecampaign.com/security/

- Adobe adobe.com, http://helpx.adobe.com/security/alertus.html
- Agoravoting.com agoravoting.com, https://agoravoting.com/misc/page/security
- androidfreeapp http://www.androidfreeapp.net, http://www.androidfreeapp.net/security-researcher-acknowledgments/
- Amazon Web Services aws.amazon.com, http://aws.amazon.com/security/vulnerability-reporting/
- Beanstalk, http://beanstalkapp.com, urlhttp://support.beanstalkapp.com/customer/portal/articles/1317175-responsible-disclosure-policy
- Basecamp basecamp.com, https://basecamp.com/security/response
- CoinBase, https://coinbase.com, https://coinbase.com/whitehat
- CoinDrawer, https://www.coindrawer.com/, https://www.coindrawer.com/whitehat/
- CryptoCat, https://crypto.cat, https://crypto.cat/bughunt/
- Dropbox, http://www.dropbox.com, https://www.dropbox.com/special_thanks
- Ebay, http://ebay.com, http://pages.ebay.com/securitycenter/Researchers.html
- Etsy, http://www.etsy.com, http://www.etsy.com/help/article/2463
- Evernote, http://evernote.com, https://evernote.com/security/
- Facebook https://www.facebook.com/whitehat/, facebook.com
- flickr http://flickr.com, https://hackerone.com/yahoo?
- FourSquare http://foursquare.com, https://foursquare.com/about/security
- GetBase, https://getbase.com, https://getbase.com/security/
- Github, https://github.com, https://help.github.com/articles/responsible-disclosure-of-security-vulnerabilities
- gliph, https://gli.ph, https://gli.ph/security.html
- Microsoft Office 365, urlhttp://technet.microsoft.com/en-us/office365, http://technet.microsoft.com/en-us/security/cc308589
- Netflix http://netflix.com, https://support.netflix.com/en/node/6657#gsc.tab=0
- prezi http://prezi.com, http://prezi.com/bugbounty/
- ShareLatex https://www.sharelatex.com/security, https://www.sharelatex.com

- spotify http://www.spotify.com, https://www.spotify.com/us/about-us/contact/report-security-issues/
- blogger http://blogger.com, http://www.google.com/about/appsecurity/reward-program/
- gmail http://gmail.com, http://www.google.com/about/appsecurity/reward-program/
- Google Search http://google.com, http://www.google.com/about/appsecurity/reward-program/
- Google Scholar http://scholar.google.com/, http://www.google.com/about/appsecurity/reward-program/
- orkut http://orkut.com, http://www.google.com/about/appsecurity/reward-program/
- yahoo http://yahoo.com, https://hackerone.com/yahoo?
- youtube http://youtube.com, http://www.google.com/about/appsecurity/reward-program/
- zynga http://zynga.com, http://company.zynga.com/security/whitehats

4 First deliverable: Responsible disclosure policy analysis.

Your first deliverable is **typed up PDF, submitted as "final project" using websubmit** containing the following information:

1. Your group members. You may have 2 or 3 people in your group; if you are a CS masters student and you want this project to count as your Master's Project, you should indicate this on your submission and work alone.
2. The name and url of the main page of the website you will analyze.
3. A link to the site's responsible disclosure policy or bug-bounty program.
4. A 1-3 paragraphs describing the process for disclosing a vulnerability that you might have found on that site, according to the site's responsible policy/bug bounty program. Make sure you describe (1) how you would go about disclosing the vulnerability, (2) what the site commits to do once the vulnerability has been disclosed to them (for example, to fix the vulnerability within X days, to not take actions against the person disclosing the vulnerability, to disclose the vulnerability on their website or blog, etc.). If you feel like the responsible disclosure policy contains any loopholes that can be used against the security researcher, you should also discuss this in your paragraph. All statements you make should be supported using direct quotes from the site's terms of service, responsible disclosure policy, bug bounty program, etc.

5 Analysis.

Your task is to prepare **(1) a poster, and (2) a 3 to 5 page report** the summarizes what information the ISA can glean from the website, using each of the four snooping techniques discussed in Section 2; you should focus especially on the first technique – eavesdropping on unencrypted network traffic – but you can also discuss the others (especially if you are studying a site that is mainly served by https!).

This is an open ended project, so, **!!!as long as you follow the rules in Section 5.1!!!**, you poster can focus on any issues in Section 5.2, or other issues that you find interesting. As usual in this class, we will be looking for in-depth research, rather than superficial information.

Poster templates, and information about printing posters, are available here:

https://cs-wiki.bu.edu/tiki-index.php?page=Printing+Posters

Posters will be presented to the class and department on May 1, at our class poster session. Your are welcome to invite colleagues and friends to the poster session. During the poster session, the course staff will go around and discuss the posters with each group, and a portion of each group's grade will be based on their ability to answer the instructors questions.

5.1 The rules.

IMPORTANT!	In the process of your analysis you may discover surprising results and vulnerabilities. Do not discuss them with anyone outside of your group without first consulting with Professor Goldberg. Failure to do this will result in an automatic F.

As this task involves interacting with a private party's computer systems it is very important that you avoid anything outside of what a normal user would do. That means, **you can look but you can't touch.** The following rules are guidelines for what we consider a "normal user" would do. But, in doing this lab, you should err on the side of caution. Just because the rules do not say not to do something, this does **not** mean that you can do it.[1]

1. DO: Use the site as a normal user would, follow links, click buttons, interact.

2. DO: Watch and record what actions the site takes, what it saves to your disk, what URLs it requests, what information it asks from the user.

3. DO: Analyze what data the site has about a user and how it saves this data. Is everything stored on the server, or is some of the data recorded locally in cookies? What sorts of vulnerabilities might this create?

[1] In Airbud a dog is allowed to play basketball because "there is no rule on the books saying that a Dog cant́ play basketball", we will not accept such an argument. The absence of a rule forbidding a particular action does not imply that it is allowed or condone it in anyway.

4. DO: Read the responsible disclosure policy for your web applications and make sure not to violate it.
5. DO NOT: Edit URLS.[2],
6. DO NOT: Change cookies, post javascript or strange characters into forms.
7. DO NOT: Do any thing which violates the law, other users' privacy, the user agreement of the web application, or the code of computing ethics and of Boston University.
8. DO NOT: Attempt to attack the server or client in any way, including but not limited to XSS, CSRF and SQL injection.
9. DO NOT: Save the webpage to disk, and then alter it and load it.
10. DO NOT: Post online or discuss your results with anyone outside your group without first consulting with Professor Goldberg.

5.2 Things to Look For

1. Cookies: Does the site use cookies, if so what data is stored in the cookies? Who can access these cookies? Are these Secure Cookies? Persistent or Session cookies? What could you learn about a user if you had access to their cookies?
2. Are ads shown to the user? Where are these ads loaded from? Who can display these ads? Can the ads contain javascript? What can an advertiser learn about the user?
3. What URLs are fetched when the site loads? Are any of these URLs offsite? Are these URLs protected with HTTPS? What are these URLs used for and what are the privacy risks? Do these URLs contain any sensitive data?
4. Does the site used mixed HTTP and HTTPS content?
5. How does the site track the user? Cache? Flash cookies?
6. Are there any side channels? For instance does the site make a request each time a user hits a key? Even if these request was encrypted what could an eavesdropper learn?
7. What plug-ins are loaded? Java applets? Flash? ActiveX? Unity? How could these be used to violate the users privacy?
8. What is the web application privacy policy? Who do they share data with? Could ISA purchase the user data from the web application's company or affiliate?
9. Does the web application violate any user privacy laws?

[2] This may seem harmless, but sometimes it is not. We have seen many examples in class where editing URLs can result in successful SQL or XSS attacks. There have also been instances of production systems being crashed by a user deleting a single field from a URL and requesting it. At least one person has been sentenced to more than 3 years in federal prison for generating malicious URLs and accessing them (weev).

5.3 The Chrome browser development tools.

Your main tool for this project will be the Chrome browser's developer tools:

https://developers.google.com/chrome-developer-tools/

or Firefox's browser developer tools:
https://developer.mozilla.org/en-US/docs/Tools

Ethan will also be holding a discussion section to cover using how to use these tools. You can access these tools by right clicking on a page and selecting "Inspect Element"; this will open an interface that allows you inspect the requests, cookies, network traffic, etc. **You must not not use the CONSOLE tab as part of your analysis.**