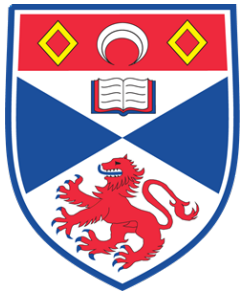
University of St Andrews

CS4203 Coursework 2

Security Tools

Author:
150008022

November 26, 2018



Part I: 1377 Words, Part II: 1007 Words,

Goal

The goal of this practical was to use and evaluate common tools and techniques used in security: steganography and penetration testing.

Part I
Steganography

Least Significant Bit (LSB) steganography - specifically LSB1 - involves replacing the last bit of each byte of an image file (called the cover image) with a bit from the data to be hidden. For example, if each character in the message requires a single byte, then it can be hidden in 8 bytes of the cover image. If we instead use the two least significant bits, we can half the image size needed to mask our message, but this can result in more obvious distortion in the original image. The message being hidden doesn't necessarily have to be text either, images can be masked using other images as well.

Experiment 1 compares various images after LSB steganography is applied to determine the effectiveness of spatial frequency as a metric for choosing cover images. In experiment 2, the effectiveness of peak signal-to-noise ratio (PSNR) as a metric is also evaluated.

1 Spatial Frequencies

1.1 Introduction

Images can contain a range of spatial frequencies. High frequency images show fine details, whilst low frequency images consist of smoother transitions between dark and light (See figure 1). In this experiment, the effect of LSB on images of varying spatial frequencies is compared, using an online image steganography tool [1].

(a) High spatial frequency image (b) Low spatial frequency image

Figure 1: Comparison of spatial frequencies in images [2]

1.2 Aims

The aims of this experiment are:

1. to determine if spatial frequency is useful as a metric for choosing an effective cover/hidden image combination.

2. to determine whether a high, medium, or low frequency image is a better candidate for a cover file.

3. to determine which combination of high or low frequency cover image and high or low frequency hidden image provides the best concealment.

1.3 Methods

In order to reduce the number of variables in this experiment and simplify masking one image in another, the images used were all cropped and scaled to the size of 640×396 pixels. Three images were used that contained either low, medium or high values spatial frequency components. The images were viewed on a 15.6" Matte FHD LED IPS display with a resolution of 1920×1080. The *Eye of GNOME* application was used to open and display the image files

A Matlab® script was written (*showDCT.m*) that converted the images to grayscale and used Discrete Cosine Transform to show the distribution of spatial frequencies. The images used and their spatial frequencies are shown in figures 2 to 6. The DCT plots show the lowest frequencies in the top left and the highest frequencies in the bottom right.

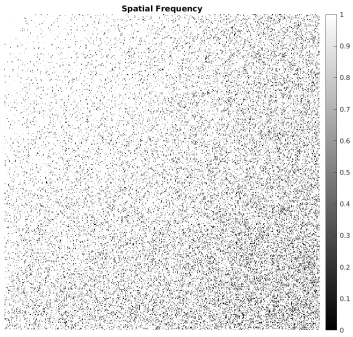


Figure 2: "Colourful": High frequency image (left) and spatial frequency distribution (right).

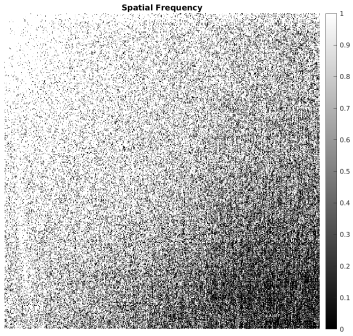


Figure 3: "Viaduct": Mid frequency image (left) and spatial frequency distribution (right).

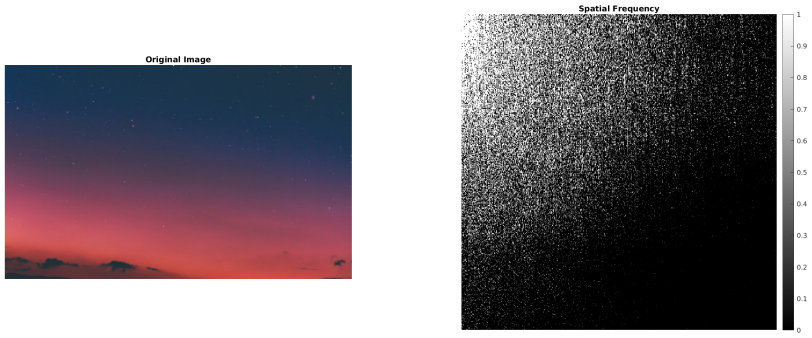


Figure 4: "Sky": Low frequency image [3] (left) and spatial frequency distribution (right)

.

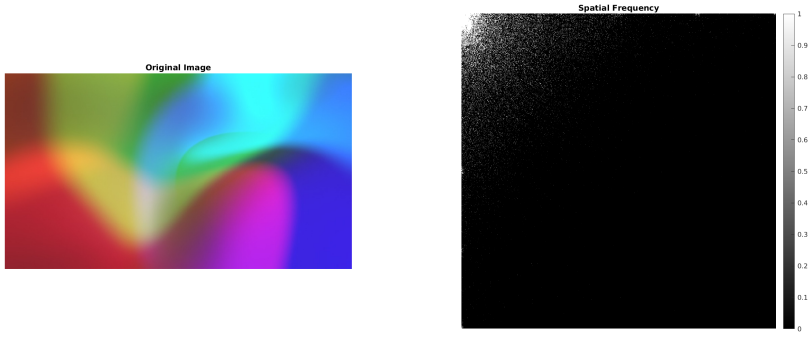


Figure 5: "Blurry": Low frequency image (left) and spatial frequency distribution (right)

.

Figure 6: "Galette": High frequency image (left) and spatial frequency distribution (right)

.

Using the online image steganography tool, each of three images (Figure 2 to 4) were combined, using the first, second, and fourth lowest bit planes. The two high frequency images (Figure 2 and Figure 6) and the two low frequency images (Figures 4 and 5) were also merged.

For aim 1, spatial frequency could be deemed useful if there is found to be some correlation between the results and the spatial frequencies of the images. For aim 2, the best cover file would be the one that after hiding have the least visible changes compared to the original. The same metric could be used for aim 3. The cover image with another hidden would first be compared to the original image, and then to the hidden image. If artefacts can be spotted when compared with the original, then the cover image offers very weak protection, but if it can only be noticed when the hidden image is also shown, then the cover image offers some protection. If neither, then the cover image is very effective.

1.4 Experimental Trials

The resuls are shown in the style of a Cayley table, where the horizontal headers correspond to the cover image used, and the vertical headers correspond to the image that was hidden. A score of 1 to 10 is used for each combina-

tion, where 1 shows clearly the hidden image, and 10 is indecernible from the original cover image. An asterisk is used where no data was recorded.

1.4.1 LSB1

Hidden \ Cover	Colourful	Viaduct	Sky
Colourful	*	10	7
Viaduct	10	*	6
Sky	10	10	*
Blurry	*	*	10
Galette	10	*	*

1.4.2 LSB2

Hidden \ Cover	Colourful	Viaduct	Sky
Colourful	*	8	4
Viaduct	10	*	3
Sky	10	9	*
Blurry	*	*	9
Galette	10	*	*

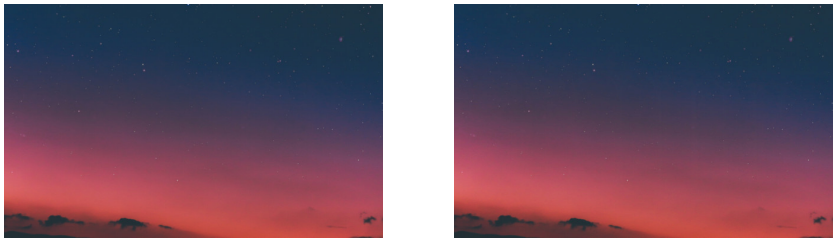
1.4.3 LSB4

Hidden \ Cover	Colourful	Viaduct	Sky
Colourful	*	3	2
Viaduct	7	*	2
Sky	10	5	*
Blurry	*	*	4
Galette	8	*	*

1.5 Outcomes

For LSB1, the following notes were taken:

- Top portion of Sky image gained unnatrual textures when colourful was the hidden image.
- Vertical columns in viaduct were faintly visible when hidden under sky image (Figure 7).



(a) Original Sky Image (b) Viaduct hidden in Sky Image.

Figure 7: Sky before and after viaduct was hidden using first least significant bit plane.

For LSB2:

- Viaduct image with colourful image hidden showed circle bright spots in sky. Colour transitions became more sudden (posterization).
- Viaudct with sky hidden also suffered from posterization
- Sky with viaduct and colourful hidden begins to clearly show shapes in hidden images.

And finally for LSB3:

- Colourful with viaduct hidden only hints at hidden image in low frequency section (dark wall background).
- Colourful image with Galette appears brighter, slight texture in low frequency regions.
- Viaduct with colourful definitely affected by posterization (Figure 8). Shows some textures from Colourful image in top portion, though rest is fairly unaffected.

The results in regard to the aims of the experiment are:

(a) Original Viaduct Image

(b) Viaduct image with Colourful image hidden.

Figure 8: Viaduct before and after having Colourful hidden within using four least significant bit planes.

1. Since higher spatial frequencies make artifacts harder to notice in an image, spatial frequency could be useful to determine if an image is effective as a cover image.

2. A highly detailed image is best suited to the purpose of cover image.

3. Hiding a low frequency image in a high frequency cover image provides the most concealement

1.6 Discussion

From the results, the colourful high frequency image showed the least distortion when used as a cover image. Only areas of less detail (blocks of solid colour) would show hints of the images underneath. The effectiveness of the less detailed image (sky) as a cover image drops sharply as the number of least significant bits used increases, even when being used to camoflauge an image with similar spatial frequencies. From the results, it appears likely that a "white noise" image like television static would be most effective as a cover image.

The experiment could be improved by using a more standardised scoring method, such as the single stimulus continuous quality evaluation (SSCQE) [4], as well as using more participants.

2 PSNR as a Metric

2.1 Introduction

Often the peak signal-to-noise ratio (PSNR) is used as a metric of image quality during compression. It can also be used to quantify the difference between two images [5]. The higher the value of PSNR for the stego image, the more similar it is to the original cover image, which suggests that humans will not perceive that any image is hidden. PSNR is used for this purpose very often as it is fast to compute and trivial to implement [6]. However, using the literal difference between pixels values has been argued to be ineffective, since steganography is most effective when the two images are perceptually similiar rather than similar at the byte level (which PSNR evaluates).

2.2 Aims

The aim of this experiment is:

1. To determine the usefulness of PSNR as metric when choosing a cover image for LSB2 steganography.

2.3 Methods

A Matlab script (*calcPSNR.m*) was used to calculate the value of PSNR between the luminance channels of two images. The luminance channel was used as this is the aspect of images that the human eye is most sensitive to [7]. This involves converting both images to YCbCr, and calculating the peak signal-to-noise ratio of the luminance (Y) channel of both images using

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{peakval}^2}{\text{MSE}} \right) \tag{1}$$

where peakval is the max value of the channel (255 for an 8-bit luminance channel) and MSE is the mean square error [8].

Using the same images from experiment 1, the value of PSNR was calculated between the stego image and the cover image. If a low PSNR value corresponds to an ineffective stego image when plotted, this correlation can be confirmed.

2.4 Experimental Trials

The results are plotted in figure 9. The script *psnrForAllLSB.m* was run for each original image to produce *test.csv*, to which the perceptual scores were then added manually.

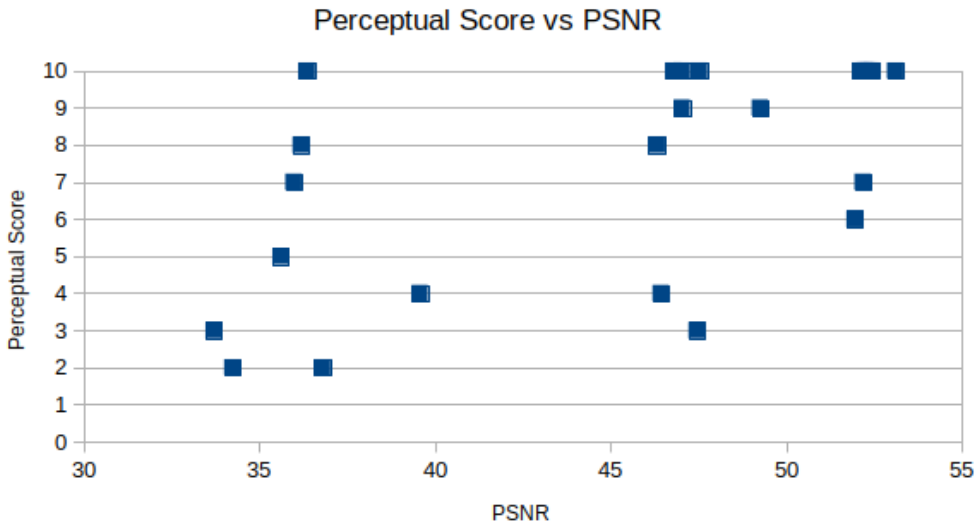


Figure 9: Perceptual scores from experiment 1 against PSNR values of cover image and the stego image.

2.5 Outcomes

No clear correlation between peak signal-to-noise ratio and the perceptual quality score can be seen.

2.6 Discussion

The experiment was carried out on very few images, and relies on the result of the previous experiment which already sufferred from few numbers and a non-standardized scoring method. Nonetheless, the results do not seem to suggest any correlation between PSNR and perceptual quality of the stego image, which agrees with other experiments [6]. Having realised this, most

literature on modern steganography favours the use of the structural similarity index (SSIM) as a metric for how similar two images are in terms of human perception [9].

Part II
Penetration Testing

The tools and processes described in Broad and Binder [10] were utilised for this part.

1 Reconnaissance

The online tool DNSInspect provided a warning that all name servers for the site are located on a single C class network [11]. This is a possible single point of failure which can be mitigated by spreading the name servers both geologically and topologically.

Whois shows that the domain name is assigned to two people in West Yorkshire (Alex Parker and Mark Ducadi), at the address Eukhost Ltd, 7 Commercial Street, Morley, West Yorkshire, LS27 8HX. This information can be incredibly useful for social engineering and other exploits, and is required by ICANN. However, many services provide protection for WhoIs information which will mask the data with generic or unrelated contact details [12].

Saving an offline copy of the site and examining the source files using the command `wget-m-p-E-k-K-np-vhttp://cybertest.uk` allowed for further examination. A few commented out lines in the `cybertest.uk.html` document provided a link to a motivating cover of the Game of Thrones theme [13], as well as a test user account `testuser@cybertest.uk` with a possible password `testunderscoreuser1`. This kind of problem can be avoided through a development cycle involving peer reviews. Once removed, the username and password would likely have to changed or removed from the system in order to avoid it being utilised by attackers in the future. Many Github repositories suffer from a similiar issue, where developers allow configuration files containing sensitive data such as passwords to be publicly available.

Another commented out section of the HTML file points us to `http://cybertest.uk/Server/AboutUs.asp`. This is an Active Server Page file,

which suggests that the ASP.NET framework has been used for developing the site.

The site does not support HTTPS, which means all connections are insecure as communications are made in plain text. Checking network requests and responses made when loading the page in Chrome, we can see that an Apache server is used to serve the website (Figure 10). This can be hidden by configuring the servers signature. Knowledge of the server type can help narrow down which attack types to use, and so obscuring this information is crucial.

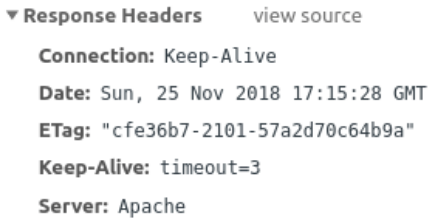


Figure 10: Response headers on GET request to *http://cybertest.uk/*

2 Scanning

In order to assess the structure of the target network further, the command `nmap-sS-T4http://cybertest.uk` was used (nmap with aggressive timing and stealth scan i.e. connections left open). A less agressive timing setting would be used if we wanted the scan to be harder to detect. The output (Figure 11) shows that several TCP ports are open. Specifially, the mail ports (pop3, smtps, pop3s, imap). There is also a mysql server running with the standard port open, which could potentially be accessed when coupled with the login details from earlier. Databases should almost never be accessible to the internet, as it is only necessary for the website server to communicate with the database, either through localhost or through a VPN. Since the database is open to the internet, brute forcing is an option.

The UDP (`-sU`) scan was then tried, though this returned no results. From this we can conclude that it is possible that the ports are open, and/or filtered by a firewall. The ACK scan (-sA) was also carried out, which returned that all ports are unfiltered.

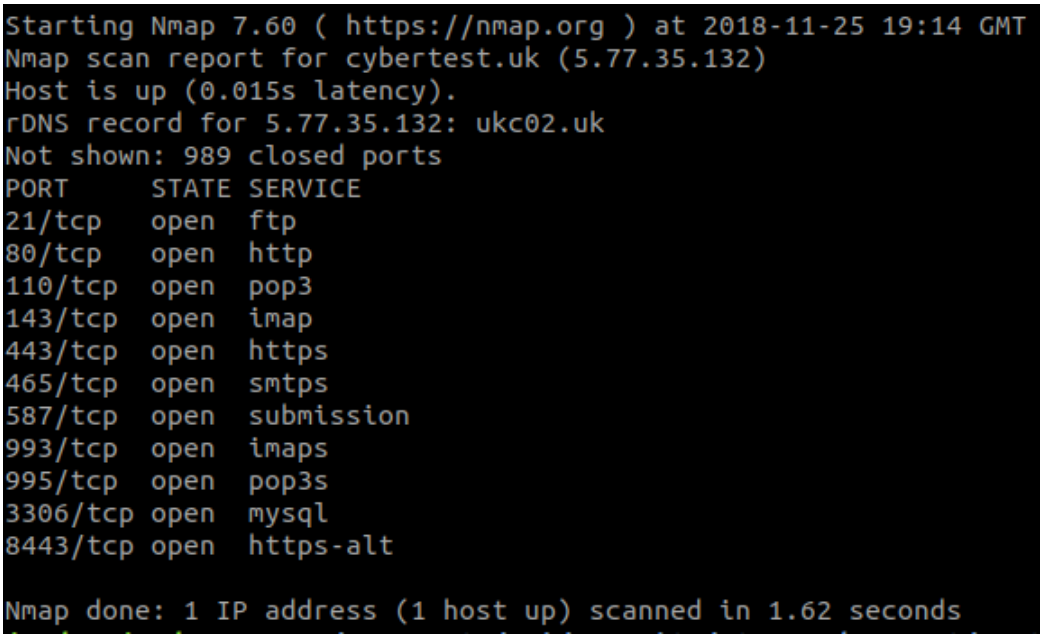


Figure 11: Response headers on GET request to http://cybertest.uk/

Port scans are difficult to protect against, and so it is best to reduce the number of attack vectors that they would present. Closing unecessary services such as HTTP on an HTTPS only web server and not running in X11 mode will reduce the number of vulnerabilities on a system. TCP wrappers can also be used to limit the amount of information available during a port scan, which will compare the requesting IP address to a list of allowed and unallowed IP addresses before determining if access to the service should be provided [14]. This mechanism also protects against IP spoofing, as a reverse DNS lookup will be used on the requesting IP address.

Not all services can be protected this way (HTTP, SMTP), and if misconfigured can provide another attack vector to be exploited. Software such as PortSentry provides TCP wrappers and other methods of protection. For example, it will redirect packets from suspect IP addresses to a different or dead system in order to hide the system being attacked.

A Metasploit project was set up to target the IP address of 5.77.35.132 (`cybertest.uk`) in order to find out more information. After scanning, it was able to determine that MySQL 5.5.57 is being used, and that the server runs on Linux 2.6.X. The last boot time from the server was also shown as being a few day prior. An overview of the scans are shown in figure 12.

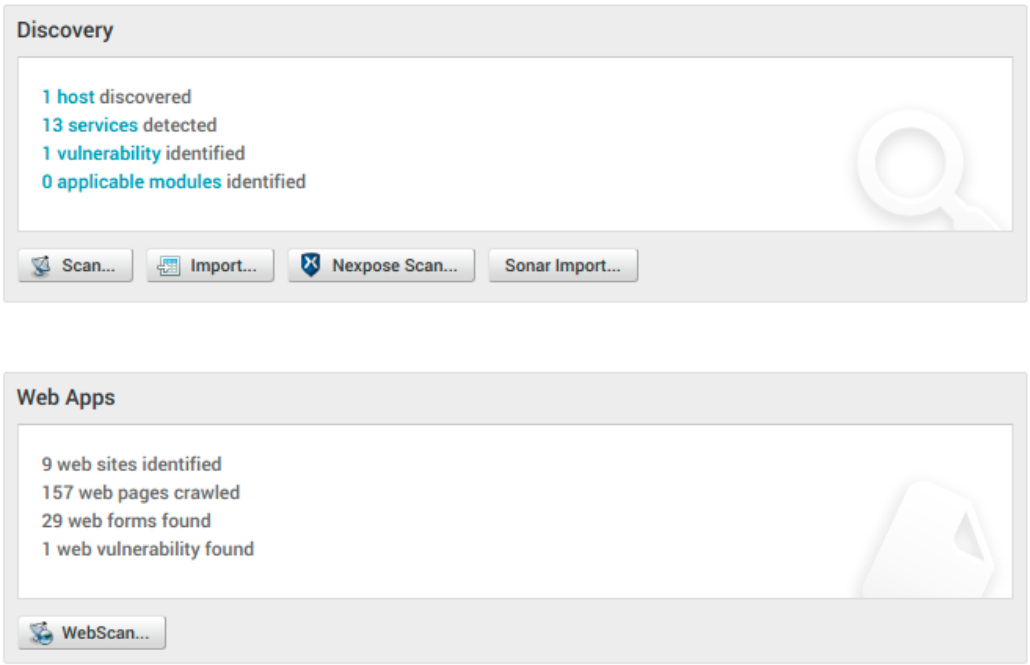


Figure 12: Overview of Metasploit scan results.

Two HTTPS ports were open. On port 8443 the login page for UKC hosting could be found. This is the portal to managing the Plesk hosting service. After using the WebScan feature on Metasploit (WMAP), it was revealed that this login page did not have an anti CSRF token. In the OWASP "Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet" [15], it specifically states that form tags using POST requests should be protected from CSRF vulnerabilties. The source for the login page at `https://ukc02.uk:8443/get_password.php` clearly features a form with method POST. This login page should preferably be hidden to the internet, and treated similiar to the database serivce, and an anti CSRF token should be used for the login form.

The reason the CSRF token is necessary is due to the fact that once the user is authenticated to the service, the browser will automatically include credentials related to the site in all subsequent requests. This allows attackers to make requests as if they were the victim. The CSRF token is a piece of information that the attackers would not be able to access, therefore can be used to confirm that the request is from the expected user. Often the exploit uses the 'src' attribute of *img* tags, as they would not be obvious to the user.

Conclusion

Both parts of this practical were enlightening and enjoyable. For part 1, reading about modern applications of steganography and its history was very interesting. Given more time, I would have like to have used the more reliable perceptual scoring methods referenced, as well as including more images in the dataset.

For part 2, the recommended reading gave a clear step-by-step guide into the process of penetration testing. Understanding the output of the tools described and the way they work has definitely helped solidify my learning for the module. I would have liked to have taken it a step further and attempted the exploitation phase, but this would have been innappropriate. Instead, I plan on experimenting against my own web-based projects in order to improve them and my own abilties in security.

Appendix

Matlab scripts, images, and result files from Part 1 can be found in directory named *stego*. Scripts were run using Matlab version 2018b.

References

[1] James Stanley. How to defeat naive image steganography. `https://incoherency.co.uk/blog/stories/image-steganography.html`, 2016.

[2] `https://www.ucalgary.ca/pip369/mod4/spatial/frequency1`, 2018.

[3] Diego PH (jdiegoph). Shooting star. `https://unsplash.com/photos/5LOhydOtTKU`, 2017.

[4] *Objective Video Quality Assessment*, chapter 41. CRC Press, 2003.

[5] Helmalatha S, U Dinesh Acharya, Renuka A, and Priya R. Kamath. A secure color image steganography in transform domain. *International*

Journal on Cryptography and Information Security (IJCIS), 3(1), March 2013.

[6] A. Almohammad and G. Ghinea. Stego image quality and the reliability of psnr. In *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, pages 215–220, July 2010.

[7] J. M. Thijssen and A. J. H. Vendrik. Differential luminance sensitivity of the human visual system. *Perception & Psychophysics*, 10(1):58–64, Jan 1971.

[8] Matlab documentation. `https://uk.mathworks.com/help/images/ref/psnr.html`. Accessed: 2018-11-24.

[9] Zhou Wang, Alan Conread Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[10] *Hacking with Kali*. Steve Elliot, 2014.

[11] Dnsinspect. `https://www.dnsinspect.com/cybertest.uk`. Accessed: 2018-11-25.

[12] Whoisguard. `https://www.namecheap.com/security/whoisguard.aspx`.

[13] 2Cellos. 2cellos - game of thrones [offical video], 2017.

[14] Roger Christopher. Port scanning techniques and the defense against them. Technical report, Sans Institute, October 2001.

[15] OWASP. Cross-site request forgery (csrf) prevention cheat sheet, 2018.