**Advanced Security 2 - Assignment 1 – 15%**

**Jake Bolger**

**C18395341**

**Part A**

Skills research

An aptitude for computers is possibly the number one quality of a cyber security expert. Other key skills and qualities include:

* In-depth knowledge of computer operating systems, hardware, and software.
* First-class problem-solving skills.
* A strong ability to work well under pressure.
* Solid telecommunications knowledge.
* Rigorous attention to detail.
* persistence and determination.
* Excellent abilities in mathematics.
* Technical Aptitude.
* Knowledge of Security Across Various Platforms. ...
* Communication Skills.
* Fundamental Computer Forensics Skills.
* A Desire to Learn.
* An Understanding of Hacking.

Certifications and Training

Below is a list of certifications and training you can complete:

1. Certified Information Systems Security Professional (CISSP):

2. Certified Information Systems Auditor (CISA):

3. Certified Information Security Manager (CISM):

4. Security+:

5. Certified Ethical Hacker (CEH):

6. GIAC Security Essentials Certification (GSEC):

7. Systems Security Certified Practitioner (SSCP):

8. CompTIA Advanced Security Practitioner (CASP+):

9. GIAC Certified Incident Handler (GCIH)

10. Offensive Security Certified Professional (OSCP):

**Discuss why there is a shortage of security personnel worldwide.**

Now a days there are a variety of reasons why there is a shortage of security personnel worldwide. These reasons are as follows: lack of corporate security programs, underfunded security resources, lack of good tools and protocols, not enough security patches, not enough email security practices and finally people who don’t follow the policies of their company in security. Other reasons are the lack of interest from younger generations, increased number of attacks and not enough skilled defenders.

**What measures/actions should be taken to address this shortage?**

The measures that should be taken to address this shortage of security personnel are, to hire managed security service providers where your in-house talent is lacking and make sure their expertise matches your compliance requirements.

Provide detailed, enticing descriptions of each role when recruiting.

Teaching students should be done through real life scenarios.

Develop apprenticeships to recruit and prepare future employees. Some states offer tax credits to employers who hire apprentices. Support unique training exercises. IBM has developed mobile cybersecurity ranges help college students and professional practice responding to attacks.

Support mental health and wellness initiatives.

The last on is invest in employee training and certification. 93% of employees would stay at a company longer if it invested in their career.

**Do you think you have enough skills to be a security expert? If no, what are you missing and if yes what are your strengths?**

No, although I have a basic understanding of security, I am no expert. I have previous experience from college including different security techniques and cyphers, encryption, decryption, and penetration testing. I am missing a lot of manual knowledge and physical experience in security. In order to become an expert, I will certainly need to get real life experience in a job or company.

**Part B**

The first static code analysis tool that I used is called ‘Guardrails’ which allows me to scan my work for errors and adjust my workflow. This was used to identify security flaws in my code from my games engines assignment I did. I installed this tool on my GitHub account and then scanned for flaws as shown below:

**GuardRails**

Graphical user interface, text, application

Description automatically generated

The flaws that were found are as follows:

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**DeepSource**

The second tool I used was a tool called ‘DeepSource’. Again I linked it with my GitHub account and used my repositories to scan for security flaws.

A screenshot of a computer

Description automatically generated with medium confidence

In the codebase report there were a lot of issues, there were 68 in total. Most of the issues were anti-pattern issues, but there were still two bug risk issues. Below is a screenshot of all the issues listed:

A screenshot of a computer

Description automatically generated with medium confidence

**Part C**

In this part I will be investigating the use of Basic and Advanced operators. For each operator I will give two examples of where they can be used. The examples will include when the operators are used alone and when they are used together or in combination.

**Basic Operators:**

**Examples of usage for + operator**

This operator is used to include keywords.

Alone: internet + security

Combination: internet + security + application

**Examples of usage for – operator**

This operator is used to exclude keywords.

Alone: application - internet

Combination: application + internet - security

**Examples of usage for ~ operator**

Used to include synonyms and similar words.

Alone: internet **~**security

Combination: internet ~security - google

**Examples of usage for . operator**

Used to include single character wildcards.

Alone: .ternet

Combination: site:cars.blogs.irishindependent.com

**Examples of usage for \* operator**

Used to include single word wildcards.

Alone: internet \* security

Combination: internet – security “hello \* world”

**Examples of usage for “” operator**

Used to include exact matches.

Alone: ”internet security”

Combination: “hello World” AROUND(3) “Jake Bolger”

**Examples of usage for | / OR operator**

Used to include keywords where either one keyword or another is matched.

Alone: internet OR security, internet | security

Combination: internet OR security ~google

**Advanced Operators:**

**Examples of usage for Allintext operator**

Only results containing *all* of the specified words somewhere on the page will be returned.

Alone: allintext:advanced security

Combination: allintext: security site:google.com

**Examples of usage for allintitle operator**

Results containing *all* of the specified words in the title tag will be returned.

Alone: allintitle:advanced security

Combination: allintitle: “operators” “for \*this assignment”

**Examples of usage for allinurl operator**

Results containing *all* of the specified words in the URL will be returned.

Alone: allinurl:advanced security

Combination: allinurl: google secuirty-keywords nikon

**Examples of usage for cache operator**

Used to search and display a version of a web page as it was shown when google crawled it.

Alone: cache:website.com

Combination: cache:website.com OR cache:google.com

**Examples of usage for Define operator**

A dictionary built into Google, basically. This will display the meaning of a word in a card-like result in the SERPs.

Alone: define:especially

Combination: define:especially + sincerely

**Examples of usage for filetype operator**

Used to limit the search to text found in a specific file type.

Alone: mysql filetype:pdf

Combination: mysql filetype:pdf – mysql filetype:txt

**Examples of usage for info operator**

Find information about a specific page.

Alone: info:security.com

Combination: info:security.com / id:security.com

**Examples of usage for intext operator**

Find pages containing a certain word (or words) somewhere in the content.

Alone: intext:security

Combination: intext: “my name is jake vs you”

**Examples of usage for Intitle operator**

Used for searching a string text within the title of a page.

Alone: intitle: index of

Combination: intitle : cars site:thesun.com intitle:earnings

**Examples of usage for inurl operator**

Used to search for a string within a URL.

Alone: inurl:mytext.txt

Combination: site:google.com -inurl:www

**Examples of usage for link operator**

Is used to search for pages that link to the requested URL.

Alone: link:www.website.com

Combination: link:thesun.com -site:irishindependent.com cars

**Examples of usage for related operator**

Find sites related to a given domain.

Alone: related:security.com

Combination: related google.com | related security.com

**Examples of usage for Site operator**

Used to limit the search query to a specific domain or web site.

Alone: site:website.com

Combination: tudublin site:website.com OR site:.gov

**Examples of usage for numrange operator**

Find results from a certain number range.

Alone: numrange:2-8

Combination: numrange:2-8 site:google.com

**Examples of usage for daterange operator**

Find results from a certain date range.

Alone: daterange: 220899-231099

Combination: daterange: 220899-231099 site: security.com

**Next, I will say whether it is possible to achieve the same results without using the operators given above.**

Yes, it is possible you can use googles advanced search which allows you to achieve the same results without using the operators above. Below is a picture and example of the advanced google search:

Graphical user interface

Description automatically generated with medium confidence

**Using the list of operators above I will identify if there are any equivalent operators that can be used in Bing.**

Operators that can be used in Bing and Google:

* Contains:
* Ext:
* Filetype:
* Intitle:
* Related:
* Site:
* \*
* -
* “ ”

Most of the Google operators perform the same task when using Bing so I have not relisted all of them. However, there are some operators that don’t work in Bing:

* Intext
* inurl
* related
* cache
* info
* allinurl
* allintitle
* link:
* ~

**Finally, I will list ten search engines outlining their advantages and disadvantages over Google or Bing.**

1. **Yahoo**

Advantages:

* Mail (1Tb) as opposed to googles 15GB
* More information.
* Search engine more powerful.
* Message boards where people can discuss ideas on any topic.

1. **AOL**

Disadvantages:

* Speed when opening is not fast enough.
* The display is not good.

1. **One Search**

Advantages:

* No cookie tracking, retargeting, or personal profiling.
* No sharing of personal data with advertisers.
* No storing of user search history.
* Unbiased, unfiltered search results.
* Encrypted search terms.

Disadvantages:

* Results can be overwhelming due to the large number of results..
* "OneSearch" does not search every item in the library even though it is named that way!
* OneSearch does not search several databases, and some databases are only partially searched.
* in other words, OneSearch is a general tool that covers all subjects, but it is not very strong in any one particular area.
* OneSearch will be less helpful if your research needs are obscure or specific.

1. **Baidu**

Advantages:

* Popular in china
* Controls 80%
* Most used internet application in china
* Baidu has a highly scalable business model,

1. **Wolfram alpha**

Advantages:

* Does calulations.
* It answers questions directly rather than pointing to another source.
* Disadvantages:
* Capabilities depend on what's in the database; not all questions can be answered.

1. **DuckDuckGo**

Advantages:

* Perfect privacy. No data on your online searches collected or stored.
* No ads targeting you based on your searches.
* No social engineering techniques are used based on your searches and other interests.
* You can be sure you are getting the same search results as all other users (no targeting or profiling).
* 1-page search results. Infinite scroll: as long as you keep going down, more search results keep loading.

1. **Yandex**

Advantages:

* With a market share of 64%, Yandex is Russia's most popular search engine.  There were 38.3 million unique visitors to yandex.ru in March.
* Besides Yandex.News, Yandex.Market, Yandex.Mail, and Yandex.Maps, Yandex also offers a number of add-on services. Location-based information can also be accessed via mobile devices.
* The Internet and mobile penetration rates are still relatively small in Russia as well.  So there is much room for growth.

1. **Firefox**

Advantages:

* It has security features that are automatically embedded into it.
* The user experience is virtually the same.
* Firefox offers a number of very helpful extensions to the browsers.
* The interface has minimalistic qualities to it.

Disadvantages:

* There are several compatibility issues.
* It consumes a lot of a computer’s memory.
* It does not automatically resume downloads.
* It struggles with HTML 5 quite a bit.

1. **Opera**

Advantages:

* Its memory and download profiles are relatively small.
* It responds quickly.
* It has several integrated protections with proven capabilities.
* There is a large community which provides strong support.

Disadvantages:

* Coding must be strictly adhered to in order for this program to work correctly.
* Companies that do web development don’t see Opera as a high priority.
* The extensions which are built into Opera are not always easy to find.
* It can offer users too many choices sometimes.

1. **Brave**

* Advantages:
* Better privacy and less tracking
* Built in ad blocking
* Faster and more efficient browsing
* Open-source software

**Part D**

1. **Injection**

Injection flaws are when an attacker uses unfiltered and often [malicious data to attack databases or directories connected to your web apps](https://owasp.org/www-community/Injection_Flaws). Two common injection attacks often get used. First, SQL injection gets used to attack your databases. Second, LDAP injection gets used to attack directories.

Injection attacks use input fields that interact with directories and databases to execute against vulnerabilities. These include usernames, passwords, and other areas that interact with the target. These fields are often left vulnerable due to the lack of an input filter when the database or directory’s development.

1. **Broken Authentication**

Authentication helps apps identify and validate users. Therefore, broken authentication can allow attackers to access and have the same permissions as the targeted user, creating severe web app vulnerabilities. Issues with authentication can give an attacker unfettered access to your data and wreak havoc on your web application.

Authentication vulnerabilities can include improperly hashed and salted passwords, leaks involving user account data, improperly set timeouts, brute force attacks,  or typical password stuffing like password1 or admin1234.

1. **Sensitive Data Exposure**

Sensitive data gets transported or stored without any encryption or other protection, leaving information vulnerable to various attacks.

There are two ways to attack unprotected data. First, while data is transported from the user to the client, attacks as a man-in-the-middle attack can be used to steal data from packets. Second, stored data, while more complicated, can be exposed through encryption keys get stored with data or weak salt/hash or passwords and credentials.

1. **Broken Access Control**

When server-side authorization is misconfigured, broken, or missing, vulnerabilities will occur that can leave your back-end open to attacks.

These attacks often happen with front-end UIs configured with components to give admins access to data or other vital app elements. In this case, most users can’t see the admin function, but those looking to find vulnerabilities will be able to uncover and exploit this flaw with malicious requests.

1. **Security Misconfiguration**

Often web applications are misconfigured, leaving an array of vulnerabilities for attackers to capitalize. Security misconfigured vulnerabilities can include unpatched flaws, unused pages, unprotected files or directories, outdated software, and running software in debug mode.

All aspects of your web applications can be affected by security misconfigurations. When a misconfiguration is found, it is vital to run a security audit to check for attacks or breaches.

1. **Cross-Site Scripting**

Cross-site scripting uses malicious code injected into benign sites to attack a user’s web browser. An attacker will insert the code through a link and, together with social engineering, will lure the user to clicking the link and executing the code. Attackers using JavaScript for XSS vulnerabilities can access a user’s webcam, location, and other sensitive data and functions.

XSS vulnerabilities are common where input is unsensitized. Additionally, XSS can allow attackers to [steal cookies from users’ browsers](https://pentest-tools.com/blog/xss-attacks-practical-scenarios/)and access browsing history and sensitive information.

1. **Insecure Direct Object References**

When database keys or files get exposed to the user, insecure direct object reference vulnerabilities exist. Because of the exposed internal objects, attackers can use enumeration attacks to access those objects and gain data or access to sensitive databases. Often authentication is either non-existent or broken.

Database objects are often vulnerable through URL parameters exposing serialized data keys an attacker can manipulate to access information. Also, static files can be manipulated and changed by an attacker to access sensitive information or other user’s data.

1. **Cross-Site Request Forgery**

Cross-site request forgeries (CSRF) use social engineering to trick authenticated users into clicking a link, as an example and take control of their sessions. Due to having authenticated sessions, the attacker can perform changes to the state of an app vs. data theft.

Applications without the proper dual authentication or cross-site tokens can be vulnerable to CSRF attacks. Those will little knowledge of social engineering are also at higher risk of their authenticated sessions hijacked.

1. **Using Components with Known Vulnerabilities**

Due diligence needs to get done when considering using a third-party code or component in your web application. Many security issues can come with using unfettered code from sources you aren’t familiar with.

To help find what components may be vulnerable, the [National Vulnerability Database](https://www.nist.gov/) has a comprehensive list of known third-party vulnerabilities to help make the best choice.

Every aspect of your app can be affected by vulnerabilities in third-party code. For example, backdoors can get added to financial services code allowing attackers access to sensitive data.

1. **Insufficient Logging & Monitoring**

Unvalidated redirects and forwards is another input manipulation vulnerability again using parameters like GET requests to execute the attacks.

An example of the vulnerability is an attacker manipulating a URL and redirecting users to a malicious site where information can get stolen using social engineering and links with malicious code or links.

**Cross Site Scripting: demo of how it works.**

**1.**

**Graphical user interface, application, Word

Description automatically generated**

**2.**

**Graphical user interface, text, application

Description automatically generated**

**3.**

**When I submit the comment it displays ‘hacked by Jake’**

**Text

Description automatically generated**

**4.**

**Using browser exploitation framework:**

[**https://github.com/beefproject/beef**](https://github.com/beefproject/beef) **download and cd into folder beef.**

**Text

Description automatically generated**

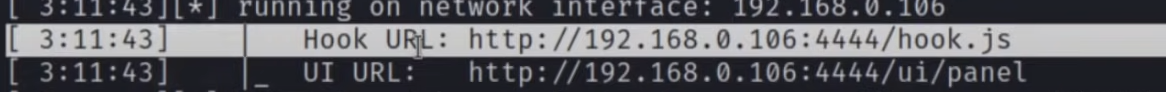
**5.**

**A screenshot of a computer

Description automatically generated with low confidence**

**Loads browser exploits framework**

**6.**

****

**7. copy paste link**

**Graphical user interface, text

Description automatically generated**

**8.**

**Graphical user interface, text, application, email

Description automatically generated**

**9.**

**Graphical user interface, text, application

Description automatically generated**

**10. link 2d interface**

**A screenshot of a computer

Description automatically generated with medium confidence**

**11.**

**Graphical user interface, application, Word

Description automatically generated**

**12.**

**Graphical user interface, application

Description automatically generated**

**13.**

**Fake pretty attack**

**Graphical user interface, application

Description automatically generated**

**14.**

**Graphical user interface, text, application, email

Description automatically generated**

**Cross Site Request Forgery: demo of how it works.**

**1.**

**Example email to take money for CSRF attack with following content:**

****

**A picture containing graphical user interface

Description automatically generated**

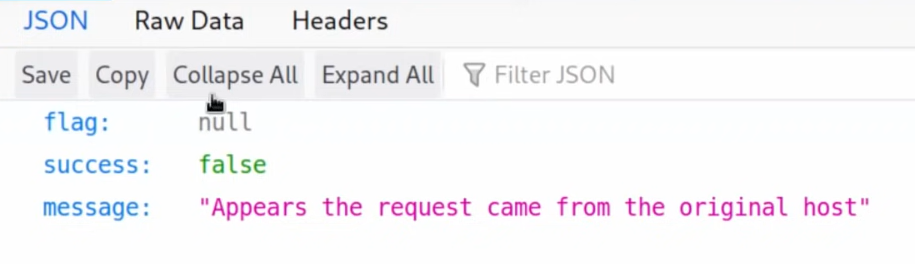
**2.**

**Get CRSF exercise:**

**Graphical user interface, text, application

Description automatically generated**

**3.**

****

**4.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**5.**

****

**Text

Description automatically generated**

**6.**

****

**Graphical user interface, application

Description automatically generated**

**7.**

**Graphical user interface, application

Description automatically generated**

**8.**

**Graphical user interface, text, application, Teams

Description automatically generated**