BITS Pilani Presentation

BITS Pilani
Pilani Campus

Jagdish Prasad
WILP

SSZG575: Ethical Hacking
Session: 11 (Webserver Exploits)

Agenda

innovate achieve lead

- What is a Web server?
- Web servers vulnerabilities
  - Vulnerabilities of Microsoft IIS/ASP/.Net
  - LAMP (Linux, Apache, MySQL, PHP)
  - IBM Websphere
  - Java Vulnerabilities

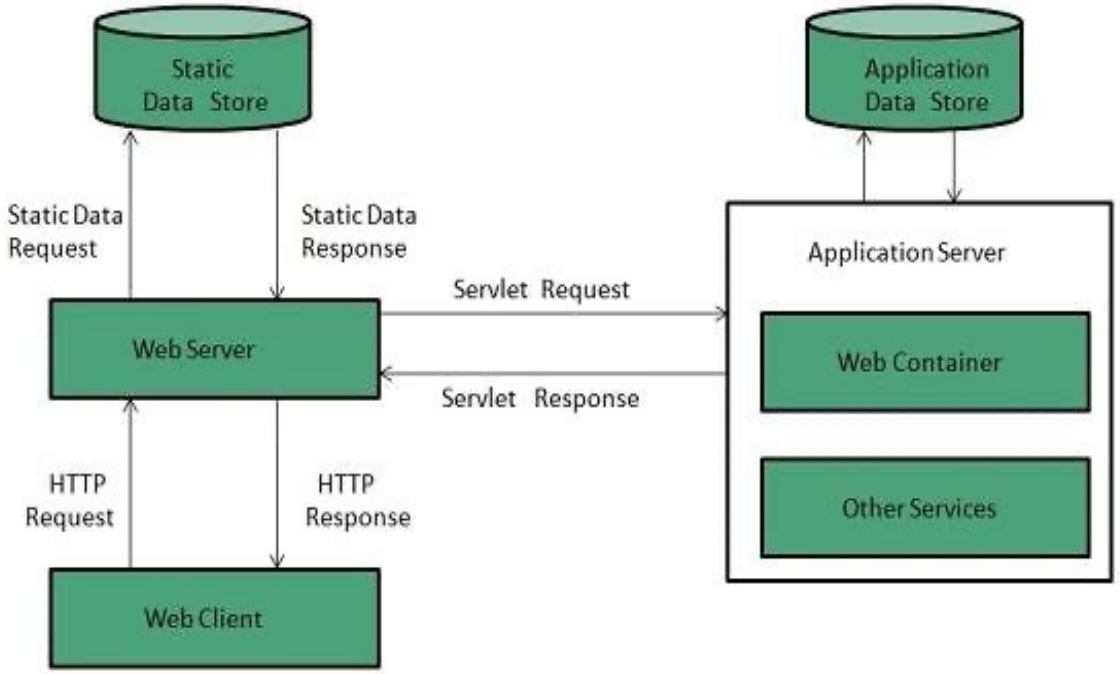
What is a Web Server?

- The term *web server* can refer to hardware or software, or both of them working together.
- Hardware:
  - A web server is a computer that stores web server software and a website's component files. (for example, HTML documents, images, CSS stylesheets, and JavaScript files)
  - A web server connects to the Internet and supports physical data interchange with other devices connected to the web.
- Software:
  - A web server includes several parts that control how web users access hosted files.
  - At a minimum, this is an *HTTP server*. An HTTP server is software that understands URLs and HTTP protocol.
  - An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

What is a Web Server?

- Web Server Process Flow:
  - Whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP.
  - When the request reaches the correct (hardware) web server, the *HTTP server* accepts the request, finds the requested document, and sends it back to the browser, also through HTTP.
  - If the server doesn't find the requested document, it returns a 404 response instead.
- Types of Web Server
  - **Static web server:** Consists of a computer with an HTTP server. Also called a "static web server" because the server sends its hosted files as-is to browser.
  - **Dynamic web server:** Consists of a static web server plus extra software like an *application server* and a *database*.
    - Called "dynamic web server" because the application server updates the hosted files before sending content to browser via the HTTP server.

How does Web Server Work?



- When client sends request for a web page, the web server search for the requested page.
- if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Popular Web Servers

- **Apache HTTP Server.** Developed by Apache Software Foundation, it is a free and open source web server for Windows, Mac OS X, Unix, Linux, Solaris and other operating systems; it needs the Apache license.
- **Microsoft Internet Information Services (IIS).** Developed by Microsoft for Microsoft platforms; it is not open sourced, but widely used.
- **Nginx.** A popular open source web server for administrators because of its light resource utilization and scalability. It can handle many concurrent sessions due to its event-driven architecture. Nginx also can be used as a proxy server and load balancer.
- **Lighttpd.** A free web server that comes with the FreeBSD operating system. It is seen as fast and secure, while consuming less CPU power.
- **Sun Java System Web Server.** A free web server from Sun Microsystems that can run on Windows, Linux and Unix. It is well-equipped to handle medium to large websites.
- Other web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers

Web Servers Security Practices

- A reverse proxy, which is designed to hide an internal server and act as an intermediary for traffic originating on an internal server
- Access restriction through processes such as limiting the web host's access to infrastructure machines or using Secure Socket Shell (SSH)
- Keeping web servers patched and up to date to help ensure the web server isn't susceptible to vulnerabilities
- Network monitoring to make sure there isn't any or unauthorized activity
- Using a firewall and SSL as firewalls can monitor HTTP traffic while having a Secure Sockets Layer (SSL) can help keep data secure.

Web Server Vulnerabilities

Web Server Vulnerabilities

- Sample files
- Source code disclosure
- Canonicalization
- Server extensions
- Input validation (Buffer Overflow, SQL injection etc)
- Denial of Service

- Incorrect configuration management plays a big role in web server vulnerability exposure.

Sample Files

- Vendors provide sample scripts and code snippets to demonstrate product features
- If poorly configured, these can leave holes in security
- Microsoft IIS4.0 came with two default files ‘showcode.asp’ and ‘codebrews.asp’
  - Files could be accessed by a remote attacker
  - Could reveal the contents of just about every other file on the server
- Latest IIS version (10.0) vulnerabilities
  - Remote code execution in Windows HTTP protocol stack
  - HTTP request smuggling
  - HTTP response splitting
- Sample files MUST be removed from production servers

Sample Files: Codebrews.asp

- Microsoft IIS 5.0 ships with a sample script that may be used to view the source code of other scripts in the sample scripts (/IISSAMPLES) directory.
- The script (CodeBrws.asp) does not adequately filter unicode representations of directory traversals. For example, an attacker can break out of the sample script directory by substituting '%c0%ae%c0%ae' for '..' in a dot-dot-slash directory traversal attack.
- This issue may be exploited to map out the directory structure of the filesystem on a host running the vulnerable script.

```
http://target/iissamples/sdk/asp/docs/CodeBrws.asp?Source=/IISSAMPLES/%c0%ae%c0%ae/default.asp
```

- The following example demonstrates that the directory structure may be mapped out using this vulnerability:
- Request: http://target/IISSamples/sdk/asp/docs/CodeBrws.asp?Source=/IISSAMPLES/%c0%ae%c0%ae/%c0%ae%c0%ae/bogus_directory/nonexistent.asp
- Response: Microsoft VBScript runtime (0x800A004C) Path not found

Source Code Disclosure

- Source code disclosure attacks allow a malicious user to view the source code of confidential application files on a vulnerable web server.
- Allows the attacker a deeper knowledge of applications and help find holes in the application.
- Attacker can combine this with other techniques to view important protected files such as /etc/passwd, global.asa etc
- Examples of source code disclosure vulnerabilities:
  - IIS: .htr vulnerability
  - Apache Tomcat and BEA WebLogic: Appending special characters to requests for Java Server Pages (JSP)

Source Code Disclosure

innovate achieve lead

- Source code disclosure techniques:
  - Using known source disclosure vulnerabilities
  - Using other vulnerabilities useful for source disclosure (such as Directory Traversal)
  - Exploiting a vulnerability in the application which may allow source disclosure
  - Exploiting detailed errors which may sometime include source code.
- **Example1:** consider a Web site running Microsoft Internet Information Server (IIS). By sending the following URL to the Web server:

  `http://www.acme-hackme.com/example.%61%73%70`

  - IIS server has a vulnerability while handling .asp files
  - If IIS is installed on a FAT partition and an attacker sends a Unicode encoded request for an .asp file (%61%73%70 is a unicode encoding of ”asp”),
  - IIS server does not recognize it as an ASP file and therefore does not execute it, but rather passes the ASP source code to the Web browser.

Source Code Disclosure

- **Example2:** by using a default sample file that comes with the IIS server, called ShowCode.asp.
- This file receives as a parameter an ASP file name and retrieves its source code.
- Specify the correct file name and directory path as a parameter in showcode.asp in the URL in the browser.
- The URL below would let an attacker view the code contained within default.asp:

```
http://www.acme-
hackme.com/msadc/Samples/SELECTOR/showcode.asp?source=/msadc
/Samples/../../../../../inetpub/wwwroot/default.asp
```

Source Code Disclosure: Example

- Web application at example.com allows users to download PDF files using hyperlinks.
- Internally it makes an HTTP GET request to a *download.php* script with a filename parameter.
- The browser sends the following request to the web server when one clicks the link:

  `http://www.example.com/download.php?filename=aboutus.pdf`The *download.php*
- Script allows users to download a specific file from the server.
- Send a request to *download.php*, passing *download.php* as the value for the filename parameter.

  `http://www.example.com/download.php?filename=download.php`
- The server-side source code of the file *download.php* is served to the browser.

Canonicalization

- Computer resources can be addressed using more than one representation.
  - File C:\text.txt may also be accessed by the syntax ..\text.txt or \\computer\C$\text.txt.
- Process of resolving a resource to a standard (canonical) name is called canonicalization.
- Applications that make security decisions based on the resource name can easily be fooled into performing unanticipated actions using so-called canonicalization attacks
- ASP::$DATA vulnerability in Microsoft's IIS was one of the first canonicalization issues in a major web platform
  - Allows the attacker to download the source code of Active Server Pages (ASP) rather than having them rendered dynamically by the IIS ASP engine
- IIS canonicalization vulnerabilities: Unicode/Double Decode vulnerabilities

Server Extensions

- A web server provides a minimum of functionality
- Extensions provide additional functionality
  - Code libraries that add on to the core HTTP engine to provide features such as dynamic script execution, security, caching etc.
- Extensions may have vulnerabilities:
  - Microsoft Indexing extension had buffer overflows
  - Microsoft Internet Printing Protocol (IPP) had buffer overflow attacks in IIS5
  - Web Distributed Authoring and Versioning (WebDAV)
  - Secure Sockets Layer (SSL) of Apache's mod_ssl had buffer overflow
  - Netscape Network Security Services Library Suite had vulnerabilities
- Microsoft WebDAV 'Translate: f' problem causes the web server to fork execution over to a vulnerable addon library when an unexpected input is sent.

Server Extensions

- Translate: f vulnerability:
  - Send a malformed HTTP GET request for a server-side executable script or related file type, such as Active Server Pages (.asp) or global.asa files.
  - These files are designed to execute on the server and are never to be rendered on the client to protect the confidentiality of programming logic, private variables etc
  - Malformed request causes IIS to send the content of such a file to the remote client rather than execute it using the appropriate scripting engine.
  - GET Command

```
GET /global.asa\ HTTP/1.0
Host: 192.168.20.10
Translate: f
[CRLF]
[CRLF]
```

Output Returned

```
D:\>type trans.txt| nc -nvv 192.168.234.41 80
(UNKNOWN) [192.168.234.41] 80 (?) open
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 23 Aug 2000 06:06:58 GMT
Content-Type: application/octet-stream
Content-Length: 2790
ETag: "0448299fcd6bf1:bea"
Last-Modified: Thu, 15 Jun 2000 19:04:30 GMT
Accept-Ranges: bytes
Cache-Control: no-cache
<!—Copyright 1999-2000 bigCompany.com -->
("ConnectionText") = "DSN=Phone;UID=superman;Password=test;"
("ConnectionText") = "DSN=Backend;UID=superman;PWD=test;"
("LDAPServer") = "LDAP://ldap.bigco.com:389"
("LDAPUserID") = "cn=Admin"
("LDAPPwd") = "password"
```

Server Extensions

- Cause for Translate: f vulnerability:
  - Arises from an issue with WebDAV, which is implemented in IIS as an ISAPI filter called httpext.dll
  - Filter interprets web requests before the core IIS engine does.
  - Translate: f header signals the WebDAV filter to handle the request but the trailing backslash confuses the filter resulting in direct sending of the request to the underlying OS.
  - Windows 2000 returns the file to the attacker's system rather than executing it on the server.

Buffer Overflow

innovate achieve lead

- Buffer overflows provides ability to execute arbitrary commands on the victim machine, typically with very high privilege levels.
- Buffer overflows types:
  - Heap based:
    - Attack an application by flooding the memory space reserved for a program
    - Difficult to execute and the less common than Stack
  - Stack based:
    - Exploit applications and programs by using what is known as a stack: memory space used to store user input.

- **Ref:** Dr. Mudge's 1995 paper "How to Write Buffer Overflows" (insecure.org/stf/mudge_buffer_overflow_tutorial.html) is excellent paper

Buffer Overflow Example

innovate achieve lead

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char buff[15];
    int pass = 0;

    printf("\n Enter the password : \n");
    gets(buff);

    if(strcmp(buff, "thegeekstuff"))
    {       printf ("\n Wrong Password \n");
    }    else
    {       printf ("\n Correct Password \n");
       pass = 1;
    }

    if(pass)
    {      /* Now Give root or admin rights to user*/
       printf ("\n Root privileges given to the user \n");
    }    return 0;
 }
```

```
$ ./bfrovrflw
Enter the password : thegeekstuff
Correct Password
Root privileges given to the user
```

Expected behaviour

```
$ ./bfrovrflw
Enter the password: hhhhhhhhhhhhhhhhhhhh
Wrong Password
Root privileges given to the user
```

Hacker entered value overrides ‘pass’ variable in memory and makes it ‘non-zero’ resulting in privilege grant

Buffer Overflow

- IIS HTR Chunked Encoding Transfer Heap Overflow vulnerability affects Microsoft IIS 4.0, 5.0, and 5.1.
  - Leads to remote denial of service or remote code execution at the IWAM_MACHINENAME privilege level
- IIS ASP Stack Overflow vulnerability affects Microsoft IIS 5.0, 5.1, and 6.0.
  - Allows an attacker to place files on the web server to execute arbitrary machine code in the context of the web server software.
  - Refer exploit details at https://www.exploit-db.com/exploits/15167
- IIS buffer overflows in the add-on Indexing Service extension (idq.dll)
  - Could be exploited by sending .ida or .idq requests to a vulnerable server
  - Resulted in the infamous Code Red worm (securityfocus.com/bid/2880).
- Apache mod_rewrite vulnerability affects all versions Apache 2.2.0 and results in remote code execution in the web server context.
- Apache_mod_ssl vulnerability (Slapper worm) affects all versions up to Apache 2.0.40 and results in remote code execution at the super-user level

Web Server Vulnerability Scanners

- There are multiple tools avails. Nikto and Nessus are two popular tools.
- Nikto
  - Performs comprehensive tests against web servers for multiple known web server vulnerabilities.
  - Can be downloaded from http://www.cirt.net/nikto2
- Nessus
  - Network vulnerability scanner that contains a large number of tests for known vulnerabilities in web server software
  - Can be downloaded from nessus.org/products/nessus/

Web Application Hacking

innovate achieve lead

- Web application hacking refers to attacks on applications.
- Finding vulnerabilities with Google.com:
  - To find unprotected admin, password and mail directories

```
"Index of /admin"
"Index of /password"
"Index of /mail"
"Index of /" +banques +filetype:xls (for France)
"Index of /" +passwd"Index of /" password.txt
```

  - To find other useful information

Search Query	Possible Result
`inurl:mrtg`	MRTG traffic analysis page for websites
`filetype:config web`	.NET web.config files
`global.asax index`	global.asax or global .asa files
`inurl:exchange inurl:finduser inurl:root`	Improperly configured Outlook Web Access (OWA) servers

Web Crawling

- Web crawling tools tools gather information about web sites like:
  - Static and dynamic pages
  - Include and other support files
  - Source code
  - Server response headers
  - Cookies
- Wget:
  - Free software package for retrieving files using the common Internet protocols: HTTP, HTTPS, and FTP
  - Non-interactive command-line tool which can be called from scripts, cron jobs, and terminals
- HTTrack/WinHTTrack:
  - A free cross-platform website copier - downloads websites and FTP sites for later offline viewing, editing, and browsing
  - Command-line version for scripting and an easy-to-use graphical interface

Microsoft IIS Vulnerabilities (1)

- HTTP request smuggling in Microsoft IIS (Jul-20)
  - Allows remote attacker to perform HTTP request smuggling attack
  - The vulnerability exists due to the way that HTTP proxies (front-end) and web servers (back-end) that do not strictly adhere to RFC standards handle sequences of HTTP requests received from multiple sources
  - A remote attacker can send a specially crafted request to a targeted IIS Server, perform HTTP request smuggling attack and modify responses or retrieve information from another user's HTTP session
  - Example

```
POST /home HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 62
Transfer-Encoding: chunked

0

GET /admin HTTP/1.1
Host: vulnerable-website.com
Foo: xGET /home HTTP/1.1
Host: vulnerable-website.com
```

Ref: https://portswigger.net/web-security/request-smuggling

Request Smuggling

- Attacker causes part of their front-end request to be interpreted by the back-end server as the start of the next request.
- It is prepended to be next request and can interfere with the way application processes that request. This is a request smuggling attack.
- HTTP request smuggling vulnerabilities arise because the HTTP specification provides two different ways to specify where a request ends: the Content-Length header and the Transfer-Encoding header.
- It is possible for a single message to use both methods at once, such that they conflict with each other.
- The HTTP specification attempts to prevent this problem by stating that if both the Content-Length and Transfer-Encoding headers are present, then the Content-Length header should be ignored.

Microsoft IIS Vulnerabilities (2)

innovate achieve lead

- HTTP response splitting in Microsoft IIS (Mar-20)
  - The vulnerability allows a remote attacker to perform HTTP splitting attacks.
  - The vulnerability exists due to software does not correct or process HTTP request headers.
  - A remote attacker can send specially crafted HTTP request and modify the response, sent by the web server.
  - Successful exploitation of the vulnerability may allow an attacker perform cache poisoning attack.

Response Splitting

- When a browser sends a request to the server, the server response contains HTTP headers along with HTML response, *i.e.*, the actual website content.
- Between HTTP headers and HTML responses, there is a special combination of characters that separate them - carriage return and line feed or CRLF.
- Web servers use CRLF to understand when a new HTTP header starts or ends.
- An attacker inserts CRLF characters in the user input to trick a target web server into thinking that an object has been terminated and another one has started
- Example:
  - Normal display is a log file: 123.123.123.123 - 08:15 - /index.php?page=home
  - Attacker is able to inject the CRLF characters into the HTTP request he is able to change the output stream and fake the log entries.

    /index.php?page=home&%0d%0a127.0.0.1 - 08:15 - /index.php?page=home&restrictedaction=edit
  - The output is as under:
  - 123.123.123.123 - 08:15 - /index.php?page=home&
    127.0.0.1 - 08:15 - /index.php?page=home&restrictedaction=edit

Ref: https://www.netsparker.com/blog/web-security/crlf-http-header/

Microsoft IIS Vulnerabilities (3)

- Privilege escalation in Microsoft IIS (Oct-19)
  - Allows a remote attacker to escalate privileges on the system.
  - The vulnerability exists due to a boundary error when Microsoft IIS Server fails to check the length of a buffer prior to copying memory to it.
  - A remote authenticated user can use a specially crafted application to trigger memory corruption and execute arbitrary code in the context of NT AUTHORITY\system escaping the Sandbox.
  - Successful exploitation of this vulnerability may result in complete compromise of vulnerable system.

Microsoft IIS Vulnerabilities (4)

- Denial of Service in Microsoft IIS (Jun-19)
  - Allows a remote attacker to perform a denial of service (DoS) attack.
  - Vulnerability exists due to insufficient validation of user-supplied input within the filtering feature.
  - A remote attacker can send a specially crafted request to the affected Microsoft IIS server and perform a denial of service attack against pages, configured to use request filtering.
  - Affects an unknown code of the component Request Filter. The manipulation with an unknown input leads to a denial of service vulnerability
  - Request filters restrict the types of HTTP requests that IIS processes. By blocking specific HTTP requests, request filters help prevent potentially harmful requests from reaching the server.
  - Request filter module scans incoming requests and rejects requests that are unwanted based upon configured rules.
  - By default, IIS rejects requests to browse critical code segments. It also rejects requests for some file name extensions.

Microsoft IIS Vulnerabilities (5)

- XSS in Microsoft IIS (Mar-17)
  - Allows a remote attacker to perform cross-site scripting (XSS) attacks.
  - Vulnerability is caused by incorrect filtration of input data within CustomErrorModule in custerr.dll library.
  - A remote attacker can trick the victim to follow a specially crafted link and execute arbitrary HTML and script code in victim's browser in context of vulnerable website.
  - Remote attacker can potentially steal sensitive information, change appearance of the web page, perform phishing and drive-by-download attacks.
- Reason:
  - Default HTTP 500.19 error page of IIS Services fails to properly sanitize user-supplied input as rendered in the path where the Web.config file of the application or directory was attempted to be loaded.
  - Any attempt to visit an URL will trigger either an HTTP 400 response from the server or will be handled by the customErrors Web.config setting of the application.
  - If a website root hosted on IIS or any subfolder on it is located in a UNC path, it is possible to craft a special link that, upon clicked, will trigger an HTTP 500.19 error page from the server rendering the javascript or html code injected as part of the path where the Web.config file was attempted to be loaded.

IBM Websphere Remote Code Execution

- A vulnerability in IBM WebSphere could allow for remote code execution (CVE-2020-4450)
- Issue occurs when serializing an object from an untrusted source.
- This could allow for a remote attacker to execute arbitrary code on the system with a specially-crafted sequence of serialized objects.
- The issue exists due to how the IBM Websphere Application Server handles the Internet Inter-ORB Protocol.
- The vulnerability exists due to insecure input validation when processing serialized data.
- Successful exploitation of this vulnerability could allow an attacker to execute remote code in the context of the affected application.
- Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.
- Failed exploitation could result in a denial-of-service condition.

IBM Websphere Remote Code Execution

- Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.
- Failed exploitation could result in a denial-of-service condition.

Examples of Exploits

- Two of most devastating internet worms Code Red and Nimda, exploited vulnerabilities of Microsoft IIS web server.
- **Code Red** was a computer worm observed on the Internet on 15-Jul-2001.
  - Attacked computers running Microsoft's IIS web server.
  - Contains the text string "Hacked by Chinese!", which is displayed on web pages that the worm defaces.
  - One of the few worms able to run entirely in memory, leaving no files on the hard drive or any other permanent storage (although some variants did).
  - Allowed an attacker, from a remote location, to gain full system level access to any server that was running a default installation of Windows NT 4.0, Windows 2000, or Windows XP and using the Microsoft Internet Information Services (IIS) Web server software.

Examples of Exploits

- Nimda first appeared on 18-Sep-2001 and caused traffic slowdowns
  - Rippled across the Internet, spreading through four different methods, infecting computers containing Microsoft's Web server, Internet Information Server (IIS), and computer users who opened an e-mail attachment.
  - Nimda's payload was a traffic slowdown itself - that is, it does not appear to destroy files or cause harm other than the considerable time that may be lost to the slowing or loss of traffic resulting in denial-of-service and the restoring of infected systems
  - Its name (backwards for "admin") refers to an "admin.dll" file that, when run, continued to propagate the virus.

OWASP Top 10

OWASP Top 10

- **Injection**
  - Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.
  - The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
- **Broken Authentication**
  - Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
- **Sensitive Data Exposure**
  - Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII.
  - Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

OWASP Top 10

- **Cross-Site Scripting (XSS)**
  - XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript.
  - XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
- **Insecure Deserialization**
  - Insecure deserialization often leads to remote code execution.
  - Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

OWASP Top 10

- **Using Components with Known Vulnerabilities**.
  - Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application.
  - If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
  - Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
- **Insufficient Logging & Monitoring**.
  - Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.
  - Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

OWASP Top 10

- **XML External Entities (XXE)**.
  - Many older or poorly configured XML processors evaluate external entity references within XML documents.
  - External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
- **Broken Access Control**.
  - Restrictions on what authenticated users are allowed to do not properly enforced.
  - Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights etc.
- **Security Misconfiguration**.
  - Result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers and verbose error messages containing sensitive information.
  - Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

Demo

innovate achieve lead

- IIS Hacking

  https://www.youtube.com/watch?v=HrJW6Y9kHC4

  https://www.youtube.com/watch?v=_4W0WXUatiw

  https://www.youtube.com/watch?v=XdbSYNhRszE

- HTTP Request Smuggling

  https://www.youtube.com/watch?v=3tpnuzFLU8g

  https://www.youtube.com/watch?v=gzM4wWA7RFo

Thank You