BITS Pilani
Pilani Campus

BITS Pilani Presentation

Jagdish Prasad
WILP

innovate achieve lead

BITS Pilani
Pilani Campus

SSZG575: Ethical Hacking
Session No: 11 (Remote Connectivity)

Agenda

- Exploiting Web servers
  - Vulnerabilities of Microsoft IIS/ASP/.Net
  - LAMP (Linux, Apache, MySQL, PHP)
  - IBM Websphere

Web Server Exploits

Examples of Exploits

- Two of most devastating internet worms in history, Code Red and Nimda, both exploited vulnerabilities of Microsoft IIS web server.
- **Code Red** was a computer worm observed on the Internet on July 15, 2001. It attacked computers running Microsoft's IIS web server.
  - It contains the text string "Hacked by Chinese!", which is displayed on web pages that the worm defaces.
  - It is also 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).
  - Allow an attacker, from a remote location, to gain full system level access to any server that is 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

- First appearing on September 18, 2001, Nimda is a computer virus that caused traffic slowdowns as it 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 appears to be the 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 known as denial-of-serviceand the restoring of infected systems
- Its name (backwards for "admin") refers to an "admin.dll" file that, when run, continues to propagate the virus.

Web Server Vulnerabilities

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

- All cause by incorrect configuration management

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’
  - These files could be accessed by a remote attacker and could reveal the contents of just about every other file on the server
- Sample files MUST be removed from production servers

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.
- Under certain conditions, the attacker can combine this with other techniques to view important protected files such as /etc/passwd, global.asa etc
- Some of source code disclosure vulnerabilities include the IIS +.htr vulnerability and similar issues with Apache Tomcat and BEA WebLogic related to appending special characters to requests for Java Server Pages (JSP)
- These vulnerabilities have been fixed but new code/scripts should be thoroughly checked

Canonicalization

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

Canonicalization

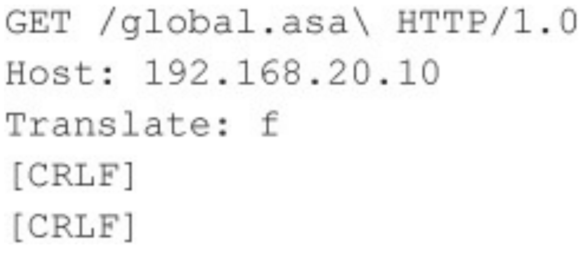
- Other most recognizable IIS canonicalization vulnerabilities are the Unicode/Double Decode vulnerabilities
-

Server Extensions

- A web server provides a minimum of functionality
- Additional whiz-bang is provided by extensions, which are 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.

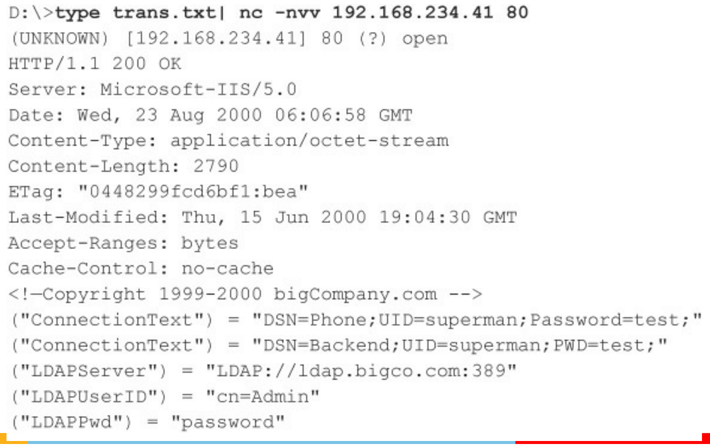
Server Extensions

- 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

- Reason 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 IISenginedoes.
  - 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

- Buffer overflows provides ability to execute arbitrary commands on the victim machine, typically with very high privilege levels.
- Dr. Mudge’s 1995 paper “How to Write Buffer Overflows” (insecure.org/stf/mudge_buffer_overflow_tutorial.html) is an excellent reference
- Aleph One’s 1996 article “Smashing the Stack for Fun and Profit,” published in Phrack Magazine, Volume 49 (phrack.com), is a paper detailing how simple the process is for overflowing a buffer.
- Buffer overflows types:
  - Stack based
  - Heap based

Buffer Overflow

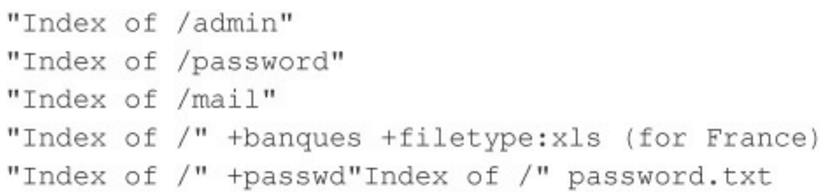
- IIS HTR Chunked Encoding Transfer Heap Overflow vulnerability affects Microsoft IIS 4.0, 5.0, and 5.1.
  - Leads to remote denial of service or remote code execution at the IWAM_MACHINENAME privilege level
- IIS ASP Stack Overflow vulnerability affects Microsoft IIS 5.0, 5.1, and 6.0.
  - Allows an 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/

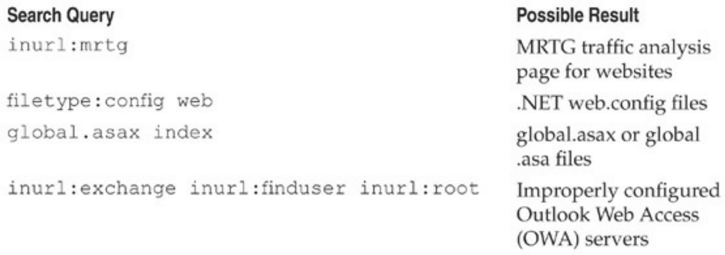
Web Application Hacking

- 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)

- 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 corrector 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 security 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 Internet Information 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.
  - Under normal conditions, any attempt to craft and visit an URL including javascript or html content on it 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

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.

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 are often 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

- HTTP Request Smuggling

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

- IIS Hacking

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

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

Thank You