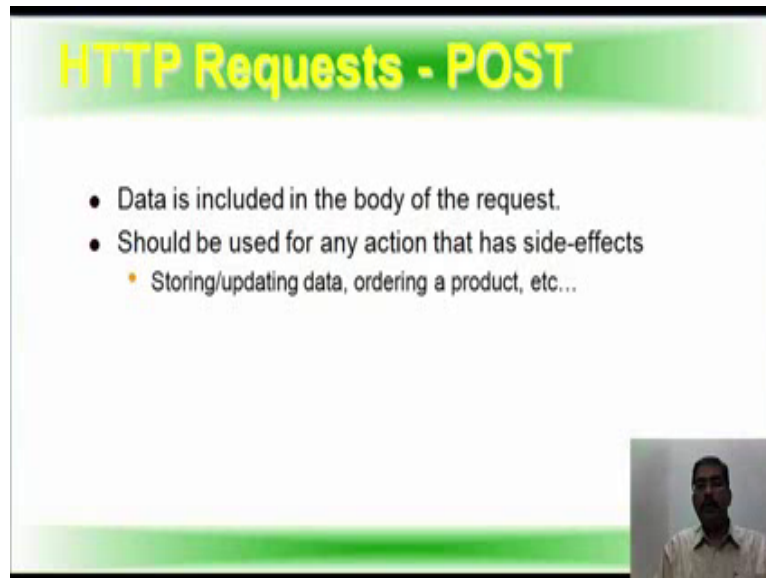


**Introduction to Information Security**  
**Prof. Dilip. Ayyar**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 59**  
**HTTP Requests-Post**

(Refer Slide Time: 00:11)

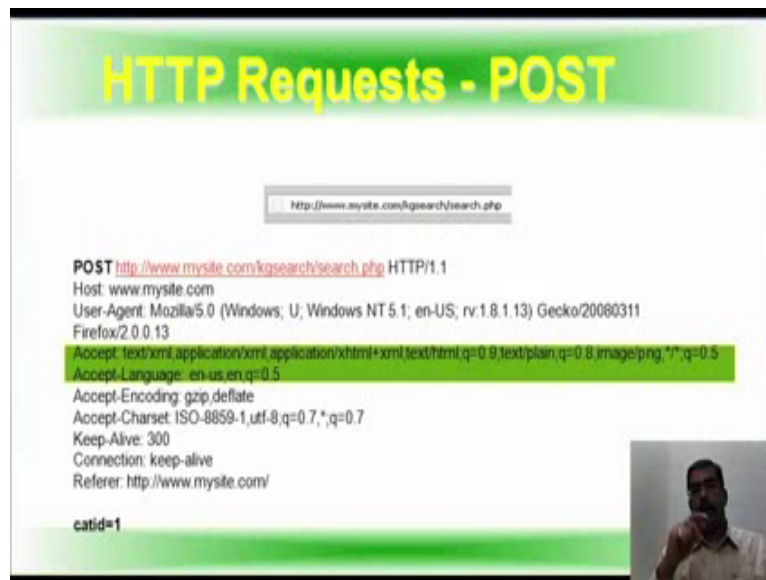


**HTTP Requests - POST**

- Data is included in the body of the request.
- Should be used for any action that has side-effects
  - Storing/updating data, ordering a product, etc...

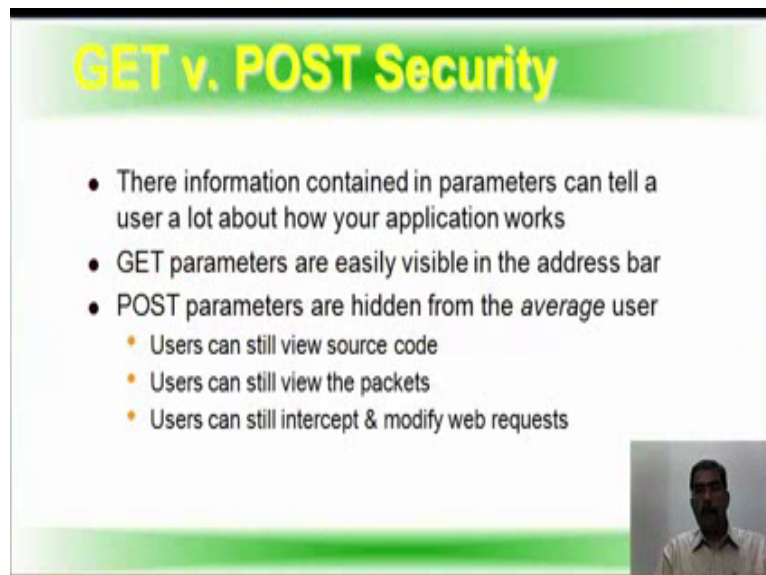
Now we will see what is the post request. Data is included in the body of the request. It should be used for any action that has side effect. It is for storing updating data ordering a product, now post request is used for that.

(Refer Slide Time: 00:29)



This is an example of a post request, same way post http mysite kgsearch search dot php. If you see the one highlighted in green, we will see; accept text or xml application xml xhtml image also it accepted. So this is how post request is send and then finally, you can see category id equal to 1, which we had shown you in the GET request also.

(Refer Slide Time: 00:58)



Now, GET vs. POST security the information contained in the parameters can tell a user of a lot about how your application work. The get parameters are easily visible in the address bar, post parameters have hidden from the average user. Users can still give the source code, users

can still give the packet, user can still intercept and modify web request. Now, to give you an example can you post or can you use that guest request to sent your user name and password.

(Refer Slide Time: 01:37)



This is another beautiful quote, Weinberg's Second Law, if builders build buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization. So I think it sums it up beautifully; take a moment read this statement and you will realize what Weinberg is trying to say.

(Refer Slide Time: 02:01)



Why web application vulnerabilities occur? So we call it a web application security gap.

Security professionals do not know the applications. Application developers and QA professionals do not know security. Now when you see on the left side as a network security professional, I do not know how my company's web applications are supposed to work so I deploy a protective solution, but I do not know if it is protecting or if it is doing what is supposed to do. Now from the application developers' perspective as an application developer I can build great features and function while meeting the deadline, but I do not know how to develop my web application with security as a feature. Now this is the security gap which is actually prevailing in the industry, there is a lack of awareness of application vulnerabilities in the security departments. Security departments scrutinized the desktops, the network and even the web servers, but the web application always escapes their measure.

Even in departments that want to audit the web application vulnerabilities. The lack of effective tools has made it practical, so as the result certification and accreditation programs rarely examine the web application. In fact, the entire development cycle is usually missing from security procedures. And this illustrates a fundamental gap between security and development, which creates these web application or which to these web application vulnerabilities. Now, many traditional information security practitioners are ill equipped to mitigate application security issues. Why, why that statement is made? Because they have little or more experience in coding that have no experience coding modern enterprise environments like say .Net or j to e that understand that there are risks, but they are not in the position to address them.

Now, if you take this specific environment or scenario in the Indian context, many of the security professionals barring though one's that have come out of engineering or master of science in cyber laws in information security, most of them traditionally were chartered accountant or finance personnel who have additionally qualified as information security auditors. Many of them are highly knowledgeable in their IT systems, but again many of them do not understand the intricacies of how the technology works that is where the gap also arises. So it becomes important that the person doing the information security audit of a web application or a web application assessment, be at least technically competent, he need not be an expert in j2e or dot net or apache or php, but at least have a ability to understand how the program works, how the programming works and where look for flaws or where to look for potential vulnerabilities in coding; that is a most important point. So it is not just a person has to be an expert an information security professional itself has to be a multi-tasking

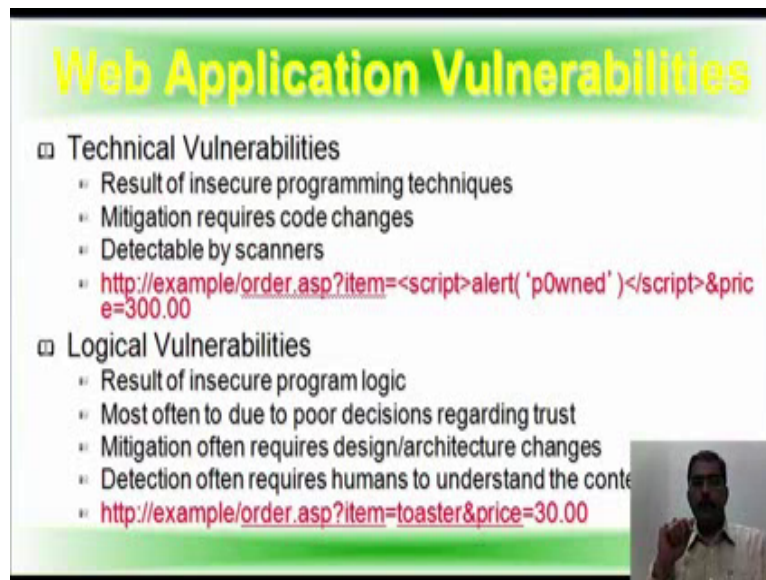
professional or multi knowledgeable professional for a simple reason that one day he will be doing a network audit, so you should know about how network work, and other day he will be doing a web app audit, the third day he will be doing an audit based on pci dss or ISO27001 or COBIT for that matter.

So, an information security professional should be up to date with the technologies with what is going on in the market about new products. Suddenly if he tells now I have web application firewall now audit the web application firewall and tell me where the configuration is gone wrong that is where the information security professional has to keep abreast with the latest technologies, the latest market developments and the needs and requirements of the client organization should be understood. The auditor should be able to fulfill it. Being overly technical is also an issue because we have to balance between the technical and administrative aspect as we have discussed in domain two also there should be a balance between administrative and technical measures; everything cannot be technical everything cannot be administrative, there should be a balance.

So, in my experience what I found is a person who is extremely good technically lacks communication skills because he is very good at working in system and getting a results, so he does not know even though he does his job properly, he does not know how to communicate to the client. How do you take the other way a person who is not at all profession unity he does an audit he knows how to communicate, but he does not know where the problems are so there should be a balance between what is required for the organization and how you present the findings to the organization. Now if there are a lot of vulnerabilities in the organization I can say, I can go to the organization and say sir you are sitting on a volcano that some of the entire process.

So, you can adopt any time meaning your systems can fail any time you will be hacked any time, but is that the way put to an organization, no so the right amount or right tone should be conveyed to organization so that is there should always be a balance between the administrative and technical measures when even in web application everything is not technical there are some aspects you will have to do manually so that is the organization realizes that importance of the work that you have done and will go on further process to secure the system.

(Refer Slide Time: 08:32)



**Web Application Vulnerabilities**

- ❑ Technical Vulnerabilities
  - Result of insecure programming techniques
  - Mitigation requires code changes
  - Detectable by scanners
  - `http://example/order.asp?item=<script>alert('p0wned')</script>&price=300.00`
- ❑ Logical Vulnerabilities
  - Result of insecure program logic
  - Most often due to poor decisions regarding trust
  - Mitigation often requires design/architecture changes
  - Detection often requires humans to understand the content
  - `http://example/order.asp?item=toaster&price=30.00`

We will see the web application vulnerabilities, there are two categories here technical and logical vulnerabilities. Technical vulnerabilities are result of insecure programming techniques. Mitigation will require code change it is detectable by scanners example is `http://example/order.asp?item=<script>alert('p0wned')</script>&price=300.00` is being write there.

Logical vulnerabilities result of insecure program logic, so you see the subject difference between the technical and logical. First is result of insecure programming technique; second under logical it is result of insecure program logic. Most of often due to poor decisions regarding trust; mitigation often required design or architectural chain so it is much more dangerous, so it requires a lot of time to be collected. Detection often requires human to understand the content. Now, example is `http://example/order.asp?item=toaster&price=30.00`. Now, to change this it is a logical vulnerability, you need to do design change for it to go properly.

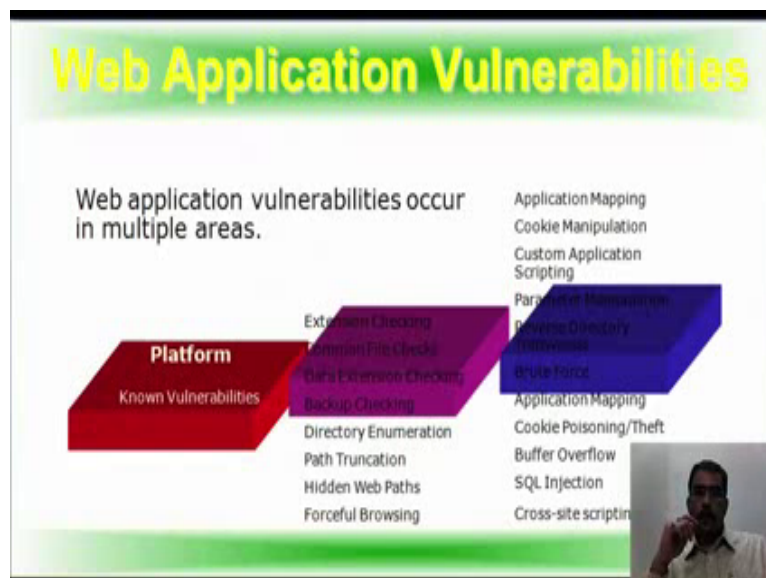
Second with importance of securing the web application also has to be an enlightened to the customer. The customer should realize that there are serious vulnerabilities and it needs to be addressed to they have to make sure that their development team understands the problem and rectifies the problem as quickly as possible that was again there will be reputational loss, there will loss of money if some financial transaction is going on. So there could be so many other issue so the organizations should perceive security as a important component in their

design overall business objective.

We will take the example of say Snap deal or Flipkart or myntra dot com, they do online transactions, so there is a lot of hits on the side the lot of financial transactions that can go on pay through credit card, pay through debit card, pay through prepaid whatever it is. So they imagine a situation if paypal or sorry if Flipkart or Snapdeal or any other online company is compromised imagine the loss of data, the loss of reputation for them. So they make sure I know for a fact because, Flipkart is very close to my office, there is a very large security teams sitting in Flipkart, who constantly monitor, who constantly test the applications to ensure that new vulnerabilities do not come.

The same of the case of backing application, they have to be doubly sure where the developer maintains proper coding standards, otherwise there could be a loss of data the same is the case with insurance industry were customer data is of immense importance. You cannot afford to lose a customer data because it is a loss of privacy. There could be other implications also or in a manufacturing unit that also there could be a problem. So you can have many examples of how these vulnerabilities will come up and why these insecure program logic and insecure program techniques are happennig.

(Refer Slide Time: 12:20)



Now, if you see here web application vulnerabilities occurring multiple areas. The first is platform it is no vulnerabilities for them then extension checking, common file checks, back up checking, directory enumeration, path truncation, hidden web paths, forceful browsing, it

comes under what category, the web server the web server itself it subjected to a variety of non-vulnerabilities all of which must readily patched for this. The right side the blue one for the web applications where application mapping, cookie manipulation custom scripting, parameter manipulation, reverse directory transversal brute force application mapping, cookie poisoning or cookie theft, buffer overflows SQL injection, cross site scripting.

Now, the actual administration and management of the content and the server is very, very important because a poorly or misconfigured server can permit system or source code disclosure; may be a file in the system can get disclosed or lost or alter or the source code disclosure itself may happen. The application itself is of at most importance; it also can inadvertently reveal the source code and system files and in some case as allow full system issues. Also it can mistakenly allow replay attacks customers it could allow a hacker to impersonate the customer till addition. It is the web application that interact is the data base to manage and track customers information and store business and transaction information. So one slip up in the web application can expose the entire system and the data base write through web browser write over port eighty, so it will dangerous.

(Refer Slide Time: 14:40)



**Web Application Vulnerabilities**

**Platform:**

- Known vulnerabilities can be exploited immediately with a minimum amount of skill or experience – “script kiddies”
- Most easily defensible of all web vulnerabilities
- MUST have streamlined patching procedures

The slide features a red 3D box on the left with the text 'Platform Known Vulnerabilities'. On the right, there is a list of three bullet points under the heading 'Platform:'. A small inset image of a person is visible in the bottom right corner of the slide.

Now, will look at platform known vulnerabilities, known vulnerabilities can be exploited immediately with a minimum amount of skill or experience like the one done by script kiddies it is most easily defensible of all the web vulnerabilities, and it must have a streamline patching process. Known vulnerabilities in the web server are obviously, a great



source of risk. The difficulty become installing patch on a lot of servers. Then they must be a streamline patching procedure when you test a patch. You see the impact of the patch on the application if everything is then you migrated to the production environment then you need take a down time in some cases because webs servers are main to be run throughout day and night twenty four bar seven without being switched off. So whether you have down time to do that that you must be easily able inventory or identify your servers for patches. Now if you miss all this you do not have to worry because if you miss a patch then a hacker will let you know that your miss the patch.

(Refer Slide Time: 16:07)

**Web Application Vulnerabilities**

- Extension Checking
- Common File Checks
- Data Extension Checking
- Backup Checking
- Directory Enumeration
- Path Truncation
- Hidden Web Paths
- Forbidden browsing

**Administration:**

- Less easily corrected than known issues
- Require increased awareness
- More than just configuration, must be aware of security flaws in actual content
- Remnant files can reveal applications and versions
- Backup files can reveal source and database connection

The slide features a purple 3D rectangular graphic on the left side and a small inset photo of a man in the bottom right corner.

Then, there are administrated issues which are less easily corrected than known issues. It requires increased awareness; it is most then just a configuration must be aware of security flaws and the actual content. Remnant files can reveal applications and versions in use; back up files can reveal source for data base connection. Now administrative issues are less easily corrupted. What does it means is that it is not as easy to fix administration issues; it requires a proper security awareness from those who manage the web site and its content on a daily basis. People who are doing on a daily basis should have proper awareness; obviously, you do not directly browsing enabled anywhere and you want the right access control list to be implemented.

It is more than just a configuration itself you must be aware of the implication of that content as well. Now for example, remnant files like readme dot txt or sample application take reveal

the applications and the versions in use. commercial applications have known vulnerabilities also. Not just the web servers and operating system. Back up files or improper application mapping can also reveal source code including the information necessary to connect to a database. Now a lot of tools which do the web application testing like your IBM appscan, commercial version, acunetix, commercial. There is core impact, there is hp web inspect so there are qualys, which is again service based model. Now, all of these report backup files, but nobody really gives importance to what that backup file means. So in this context when you see now the files can reveal the source code if that backup file contains the source code or some critical information it could be news or it could be party time for the hacker.

(Refer Slide Time: 18:24)

**Web Application Vulnerabilities**

- Application Mapping
- Cookie Manipulation
- Custom Application Scripting
- Parameter Manipulation
- Reverse Directory Transversal
- Brute Force
- Application Mapping
- Cookie Poisoning/Theft
- Buffer Overflow
- SQL Injection
- Cross Site Scripting

**Administration**

**Application Programming:**

- Common coding techniques do not necessarily include security
- Input is assumed to be valid, but not tested
- Unexamined input from a browser can inject scripts into page for replay against later visitors
- Unhandled error messages reveal application and database structures
- Unchecked database calls can be 'piggybacked' with a hacker's own call, giving direct access to business data through a web browser

Now, again when we come to application programming, the common coding techniques do not necessarily includes security. Input is assume to be a valid, but is not tested like I said of a few slides ago. Unexamined input from a browser can inject scripts into a page or replay against later visitors unhandled error messages reveal application and database structures unchecked database calls can be piggybacked with the hackers own database call giving direct access to business data through a web server. Now the application logic itself must be carefully consulted and must include security mechanism. So you should never assume that the input that you received is what your expected it must be tested, it must be validated and then it must be filtered.

The simple thing is test before you inject, ingests; and you have to be very careful on how

you call the files especially if you pull files directly from the file system you could expose your source code or even worse you can expose system files which could be very dangerous. And you have to be particularly careful to remove anything that even resembles the java script or db script. Inadvertently storing scripts entered from a hacker will allow them to be replayed against your customer resulting in your web site attacking your customers or divulging session information to the hacker. And then you have to handle all error messages properly; unhandled messages or raw messages, raw error messages are a road map through the application and the database. Another important thing a structure you have database calls carefully, and then scrutinize any user input that will become used or that will be used in the quick, so carefully constructed input could allow a hacker to piggyback your credits.