

**Information security - IV**  
**Prof. Vasan**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

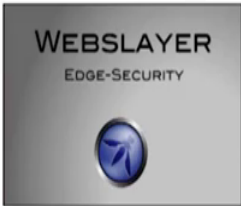
**Module - 21**  
**Lecture - 21**  
**Server Side Attacks (Contd)**  
**Tools in Kali Linux**


So, in this session, we are going to continue looking at few more tools on the sever side attacks there could be used for penetration testing. So, we actually had looked at the (Refer Time: 00:25) the tools before and in the session we are going to look at a tool called Webslayer.

(Refer Slide Time: 00:30)

[Kali Linux: Webslayer?](#)

WebSlayer is a web application brute-force tool.  
WebSlayer can be used to brute-force the Form (User/Password) ,  
GET , and POST parameters.  
WebSlayer can also be used to identify resources not linked such as  
scripts, files, directories, and so on.



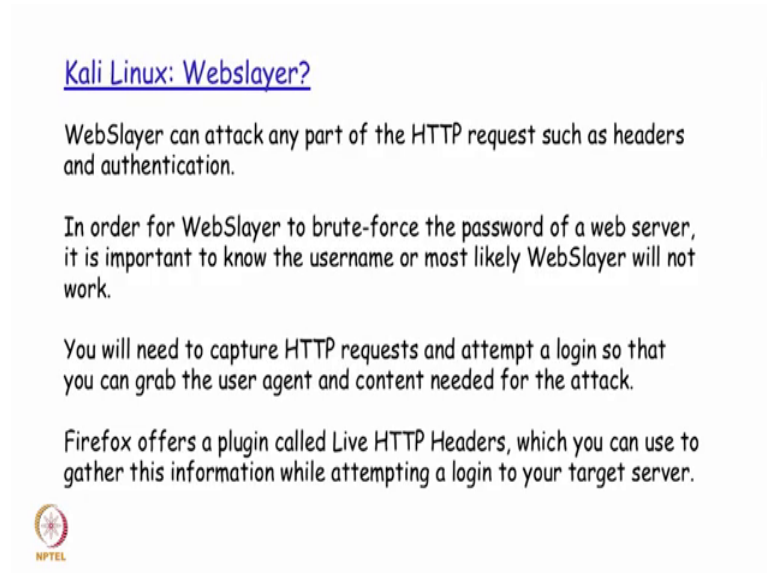


So, what exactly the Webslayer? We were discussing briefly about the brute force mechanism of actually trying to find out the credentials, user credentials that could be potentially used for entering into web server. So, web slayer is basically a brute force tool is one of the brute force tool that is actually used as a web application. So, I could actually use this tool to sort of brute force the form details like the user name, password, a combinations and also look at the get and the post parameters.

So, the tool Webslayer is not typically used standalone manner, but it is actually used along with the few plug-ins for Firefox it is actually available. So, we will actually see

them how they have got to be made use of. So, Webslayer can basically attack any part of the http request such as headers and authentication.

(Refer Slide Time: 01:31)




Kali Linux: Webslayer?

WebSlayer can attack any part of the HTTP request such as headers and authentication.

In order for WebSlayer to brute-force the password of a web server, it is important to know the username or most likely WebSlayer will not work.

You will need to capture HTTP requests and attempt a login so that you can grab the user agent and content needed for the attack.

Firefox offers a plugin called Live HTTP Headers, which you can use to gather this information while attempting a login to your target server.



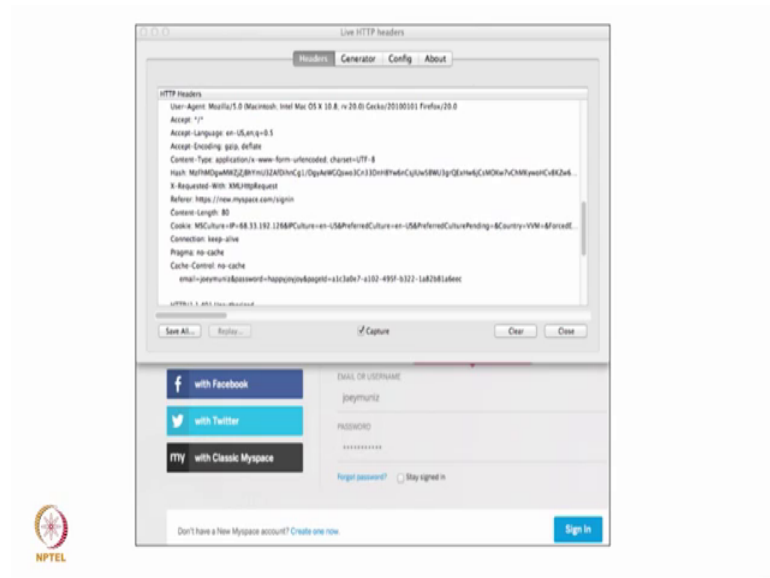
So, it could actually make use of details that is there as part of the header information, as well as details that is there as part of the authentication right. So, it could actually a brute-force password details from a standard dictionary file, to find out if it is actually also matching any standard password that could potentially be used for that particular user combination on that on that web server right.

So, first what we would actually need to do is, in order for this to actually work to the fullest possible extent, we need to capture the http request and then do a login attempt. So, that we would be able to get the user agent details, as well as the content the rest of the content needed for the attack. So, in quite of few situation the web server would actually be a differing in its behavior, depending on which browser actually try to establish the connection to it right.

So, how does the web server come to know about the browser is basically, with respect to the value that is actually filled up in the user agent field of the client request. So, in order to gather the user agent request that is actually required to be part of what we feed in as an input into the webslayer as well as may be the rest of the headers as well. We first try to do a capture of the http request to the same server from which we get these details and then feed this as an input into the webslayer tool.

So, Firefox basically offers a plug-in called Live HTTP headers, which is actually downloadable from the Firefox plug-in site which we could potentially make use of together all the information that we require for attempting a login to this particular target system right.

(Refer Slide Time: 03:39)




So, this is basically the live http headers screenshot, where even we attempted to find out what kind of headers are used when a request from the browser is going to the web server you had all the details that is presented here. So, you have a user agent, you have accept and accept language and so want right. So, for example, even you could actually get the cookie deleted details also, which is typically used by some servers to sort of allow or deny access to certain parts of their content as well as to find out what did this particular client do, when it access this particular server sometime back in the past.

(Refer Slide Time: 04:26)

### Kali Linux: Webslayer?


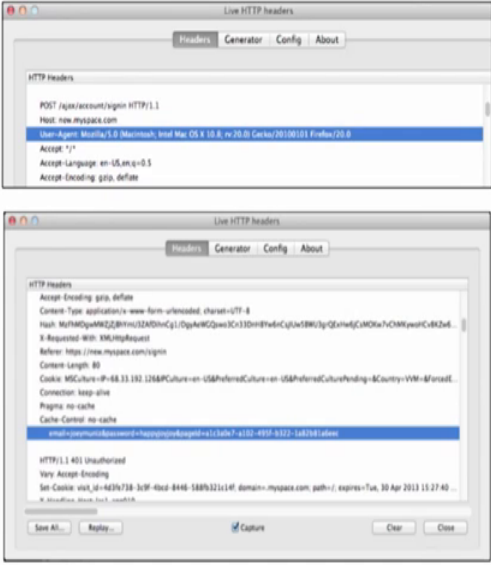
The important parts of information captured from the Live HTTP Headers used in WebSlayer are  
the User-Agent and  
Login Credentials

as shown in the following examples:



So, the most important part is the information that is actually captured from this live http plug-in is, the user agent in the login credentials.

(Refer Slide Time: 04:38)



So, the user agent field as you were discussing will basically be containing the details about the client that actually tried to initiate the attack. So, you have a Mozilla slash 5 dot o all this actually listed from which platform, the this particular client request is actually come in which browser which version of the browser, which version of the OS for example, the Mac OS that is running there and so on and so forth. So, all these

basically comes in as part of the user agent and sometimes we will also basically make user authentication fields like for example, we have the email as the authentication field here with the UID and then the password combination here right. So, depending on which site we are actually trying to get in or try to brute force enter, we might have to actually use part of these details that is being captured by this live http plug-in.

(Refer Slide Time: 05:30).


[Kali Linux: Weblayer?](#)  
In the Attack Setup tab there is an url field, which must be filled with the target URI. Below the URL field are the Headers and POST data input fields.

There is an option to set the payload type, which can be Dictionary, Range, or Payload .

The Dictionary can be a file containing payloads, which can be a custom file or selected from a list of available dictionaries.

The Range setting can be used to specify the range for the attack.

The Payload setting can import a payload from the Payload Generator tab. Lets look at an example



So, what are we going to do we going to open up the weblayer and sort of in the attack setup tab we are going to first enter the URL field. So, this URL field will be the server that we are actually trying to brute force attack with some sort of random username password combination. So, whatever is a target field target URL that we are trying to attack, will actually be entered as part of this URL field along with maybe the headers and the post data that was actually captured as part of my live http plug-in right.

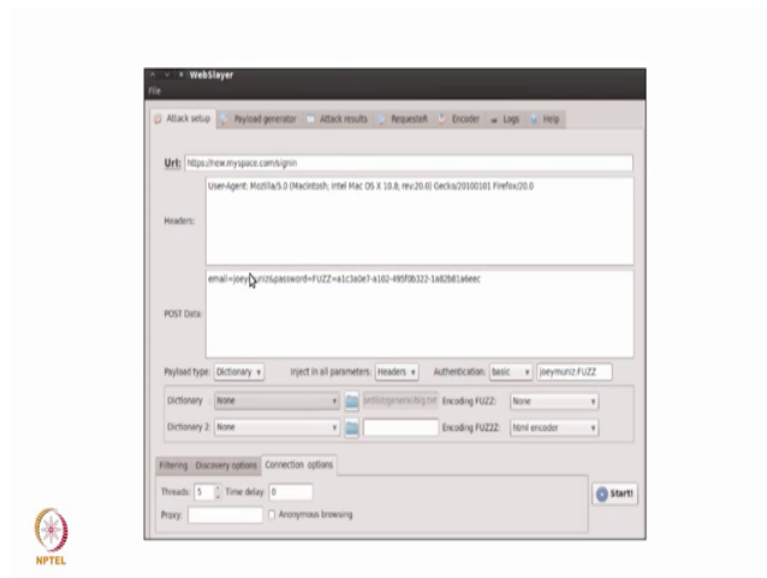
So, now I actually have an option to basically said the payload type so, which could actually be a dictionary range or a known set of payloads. So, when I say dictionary what we actually mean here is that, if we want to really tried to find out by brute force method what kind of password are actually used right. So, I could really have a set of strings in a particular file, which I could mark as a as possible passwords that needs to be attempted to be checked and then give that as a dictionary file or I could basically have the range set so that if for example, I need to let say that we are actually having the credit card as

one of the data fields in my HTML form, that is going to be submitted as part of the post request to the server right.

So, when we know credit card again I have different types of credit card that is possible. So, let us say a visa card or MasterCard or whatever it is, and all these if you strictly observe has a particular range associated with that. So, if I for example, say that I want to use a visa MasterCard range I could potentially have the range setting done from a particular starting number to a particular ending number.

At least the first the few digits of the number and put a pattern associated with that. So, that the attack will actually be done starting from that particular number and ending after that number that has been mentioned as part of the range field here and similarly I could also set the payload setting appropriately by sort of doing an import of a payload from the payload generated tab. So, we will be actually looking at the payload generated that subsequently, but that is the basically a dynamic form of generating what has to be submitted as a payload for doing this brute force attack to get the authentication details.

(Refer Slide Time: 08:11)



So, here if you see I specify the URL, whatever the URL that we are going to be actually trying to target to do a brute force attack, and you see the word first here FUZZ it is a keyword that I will basically come to. So, whatever has been the headers that has been captured from the live plug-in of my Firefox that we just saw a few slides back, I basically take those header fields and then put it in here and if at all there is any specific

post data that I want to really use slow like for example, the back the this particular weblayer tool works most of the times only if you actually give the correct username.

So, if we give the username, it basically tries to find out how it could actually doing do a brute force attack and get the corresponding password for the user name. So, suppose if we want to specify the username as part of the post field. So, we could actually mention it here.

And then if you see here, there are different options that is there like the payload type and what kind of parameters has to be injected, and what kind of authentication and so on and so forth and we see the dictionary options that is here right. Now we were talking about the keyword called first. So, the moment we say first the weblayer tool basically takes this as a keyword for into basically take the strings one by one from the specified dictionaries here, and then injected for one request by replacing the word first year.

For example, if I basically have the post data where I am specifying the username field and then I specify the password field password equal to f u z z there what it would actually do was it would basically take the strings one by one from the two dictionaries that has been actually mentioned here, whatever has been selected as part of the dropdown menu option here, and then replace that for the password field because I have specified password as first right.

But since it has actually been mentioned the this keyword first as actually will mentioned as part of URL field. So, what we are actually trying to find out here is, a valid URL field location or a list of possible URL targets, that will be available as part of this particular web application right. So, this is one example by which we could really try to find out, what kind of sub directories or sub folders are there under my web application right.


(Refer Slide Time: 10:56)

### Kali Linux: Webslayer?

The following example shows taking the login information captured in Live HTTP Headers while attempting to access myspace .

The wrong password is switched to the keyword FUZZ so that WebSlayer knows where to attempt the brute-force.

The Authentication tab has different security options for the example, the authentication is set to basic with the username joeymuniz followed by the keyword FUZZ



So, by doing it in this manner and then saying start, the tool is actually going to start working and then I m going to basically have a list of if you see here.

(Refer Slide Time: 11:00)



The screenshot shows the Webslayer application interface. The top menu includes 'Attack setup', 'Payload generator', 'Attack results', 'Requests', 'Encoder', 'Logs', and 'Help'. The main window displays a table of attack results with columns for Timer, Code, Lines, Words, Chars, MD5, Payload, Cookie, and Location. The table contains 11 rows of data. The 8th row is highlighted in blue, indicating a successful attack. Below the table, there is a browser window showing a 'Moved Permanently' message with a search bar and the text 'The document has moved here.' The status bar at the bottom of the browser window says 'Attack finished OK'.

Timer	Code	Lines	Words	Chars	MD5	Payload	Cookie	Location
1.182218	200	848	13345	138713	122d9059c838562995f39645c4d30	http://www...		
1.138819	200	848	13345	138713	2e7f9d95f51c11c137419c2121849f19	http://www...		
1.180130	200	848	13345	138713	3e0d0f9e4e93068564637180f1f1	http://www...		
1.188441	200	848	13345	138713	f507096479f189974c30548072865	http://www...		
6.280393	301	7	26	250	1796b3f841115a7c24ac85c98083e	cgi dash bin	http://www...	
4.438897	200	848	13345	138713	7796ac52c3a5054c13068830405c3	http://www...		
3.725867	200	848	13345	138713	3995b6af5519388861311eac79c795	http://www...		
4.766964	200	848	13345	138713	62294586e136c08646f7944022864	http://www...		



So, we will actually have the list of URLs and output for every attempt that it is actually maid right. So, you see the different payloads here, and then it had actually found out that for cgi dash bin as the payload from one of the files dictionary file that has been given it has actually found out to be a valid URL because of which you see here a different color associated with this right like as compared to the rest of the colors right.




(Refer Slide Time: 11:30)

[Kali Linux: Webslayer?](#)

After importing the payload into the attack scenario or selecting default dictionaries, you must select where the payload will be injected by WebSlayer.

Placing the keyword FUZZ on the URL being attacked does this. For example, the following screenshot shows the target `http://www.thesecurityblogger.com/FUZZ` in the attack URI field where FUZZ is an attack leveraging two existing dictionaries found in WebSlayer



So, another way of using this as I was telling you we could actually have the keyword first selected as part of my password field and so, whatever has been actually set as a payload or has buying up a dictionary in this particular example, the world first the keyword first will actually be replaced by different strings coming out of the payload or from the dictionary file, and then for each run the replacement of this keyword will actually happened with the string from the from the given payload as given dictionary file, and they need to let them to see if it is getting a valid response back right. So, this is another possible way by which this could potentially be made use of. So, the authentication tab has basically got different security options.

So, here in this particular case the authentication has been set as basic, but if you see the dropdown menu there, you would also see some sort of other authentication options that we could potentially make use of to do a different kind of brute force testing mechanism on the selected URL.


(Refer Slide Time: 12:37)

[Kali Linux: Webslayer?](#)

The payload generator is a tool that you can use to create custom payloads. You can load dictionaries, numeric ranges, character blocks, permutations, credit cards, usernames, and other settings.

You can concatenate and create a final payload that can be uploaded into the attack tab for a customized attack.

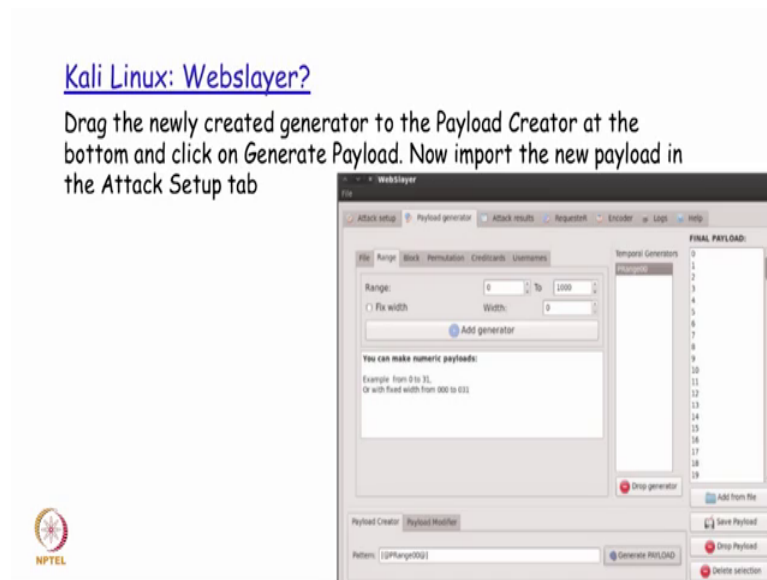
An example of defining a range payload in the Payload Generator tab can be seen in the following screenshot. The example shows setting the range payload from 0 to 1000. Once the range is selected, we click on the add generator button, which will generate a Temporal Generator.



So, coming down to the payload generator so, what exactly is a payload generator? So, the payload generator is a tool that one can make use of to create custom payloads. So, I could actually have different types of dictionaries concatenated together possibly I could have different numeric ranges I could have character block. So, if I know that there is a particular password combination, password pattern that is enforced by the server saying that it should always be starting with a character or it should always be starting with the numerical number right. I could actually appropriately have different payloads which in which I specify the pattern, saying that first digit should always be starting from 1 to 9 does a first character.

Because of the requirement the first character should be a number, it should be starting from 1 to 9 and so on and so forth right. So, I could actually have different kinds of possible permutations on this different inputs, and I could potentially give as part of my payload, and the payload generator tool is something that is going to help me to build a complete a set of payload which I could potentially concatenate all of them together, and then create one final payload that is actually uploaded as part of my attack tab attack tab in the tool, for doing customized attack.

(Refer Slide Time: 14:01)



So, if I basically look at the example here. So, there I have a tool saying I have a setting with the range for the payload generator, saying that the range should be from 0 to 1000 right. So, from 0 to 1000 I actually have the full payload that is actually generated, and either a either I could actually add this as part of my pre existing payload or I could really have this as a separate file and so on depending on my specific requirements.

So, that the values that is actually been taken as part of the payload here, would also be part of my brute force testing. To see if any of those generated payload data here as part of my specific range input that I have given is also matching with the may be that maybe the URL or maybe the password combination or whatever I am actually trying to brute force test that particular server with right.

So, with the payload generator I would actually be able to also generate a customize payload, which turned out to be very very powerful especially when the particular webs application server that we are trying to brute force test is having a very complicated password and a very highly secure password authentication mechanisms, and also sorts of enforces it mandatorily.

So, in that kind of a scenario, having a customized payload generator with different kinds of permutations and combinations, helps us to make sure that our payload that is being submitted as part of the attack is something that is very comprehensive with a very high chance of determining and being successful in the brute force testing.

So, here in this case if you also see that attack research will actually be available as parts of this particular tab finally, once a testing is completed. So, likewise you have the logs and so on. So, these different menus you could really play around with and try to get yourself familiar, because this particular tool is a tool that is very very handy and very powerful especially when we need to do brute force testing on a web application server.

Thank you.