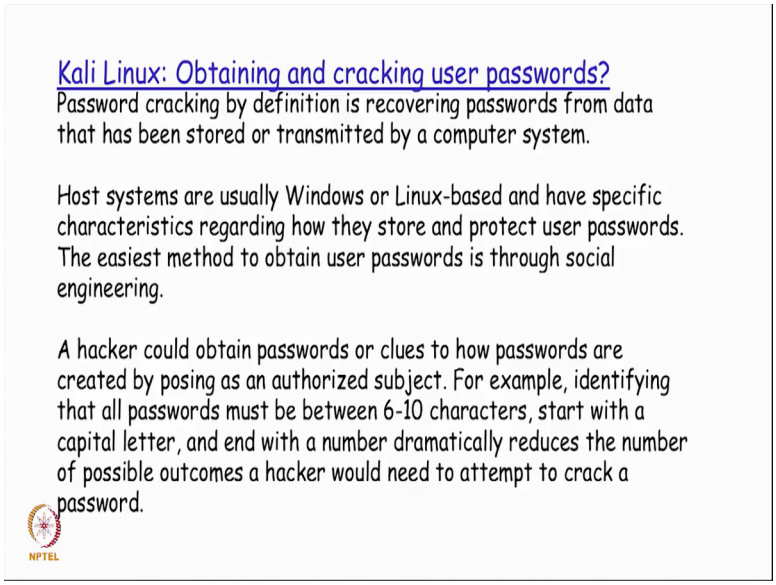
Information security - IV
Prof. Vasan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 25
Lecture – 25
Client Side Attacks (Contd)
Tools in Kali Linux

So, in this session we will actually continue more tools related to the client side attacks that could be used for penetration testing and see how that will help us in our objective of trying to do attacks from the client side. So, one of the very important things that could be required as part of the client side testing attack testing would be to see how we can actually try to crack the user passwords.

(Refer Slide Time: 00:40)



So, there was actually a tool called john the ripper that we actually seen as part of our earlier set of tools, but here we are going to look at it more from the different type of techniques that are to be followed depending on what operating system is actually running on the client. So, password cracking or determination by definition is basically trying to actually recover the passwords from the data that has been stored or transmitted by the client or the server. So, depending on which one is actually trying to get authenticated, right now.

So, most of the times if we see the host systems are typically either based out of based on any flavor of windows operating system or it will be a Linux based operating system and possibly a few of them would also be Mac based right, but any other flavor of Unix like a solar (Refer Time: 01:36) HP-UX or ax or whatever will be very very closely aligned with what we are observing as a behavior on a Linux system. So, whatever practices one deploys on for Linux system could actually be used for any other flavor of the Unix operating system also.

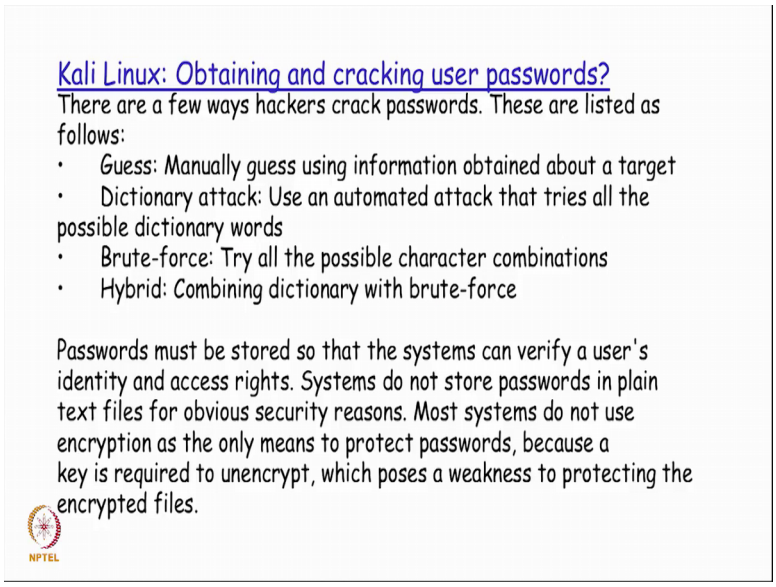
These different operating system like windows Linux and Mac OS and so on have some sort of differentiation based on how they actually store the user password as well as also protect the user password, right. The first step that potential penetration tester would actually try to deploy for trying to successfully get the user password is through social engineering. So, using any kind of an interaction over the social media or on a face to face one to one communication with the target the penetration tester could basically try to find out the kind of patterns that the passwords are supposed to be in which will give a clue to the attacker to sort of reduce the amount of data points that he will actually be trying to test it with brute force.

So, because quite a few situations what we will actually see here is that when as a legitimate user if the if the attacker basically goes to the target system and tries to a create an account or set a password or whatever it is the message coming out from the system would basically give the attacker means to find out what is the pattern that it is actually expecting, right. So, for example, as it is given here there could be a system in which all the passwords are required to be between 6 to 10 characters, right and that could be another rule saying that it should always start with the capital letter and end with the number, right.

Now, all this kind of combinations which is actually openly documented by the system what it helps the attacker is that it basically narrows down the complete data points that the attackers should use for doing a brute force attack possibly on that particular system to determine the password and crack it. These kind of initial measures are adopted generally by the attacker when they are basically set out with an objective of trying to crack the password of the targeted system.

So, as we have seen before there are a few ways the hackers could crack the passwords. So, here they could be doing a guess work or it could be a dictionary attack basically trying to run an automated program which will actually check against each of the dictionary words giving that as a target and then seeing if the authentication is successful then could be a brute force or it could be a hybrid between any of the any of the these three parameters, right.

(Refer Slide Time: 04:27)

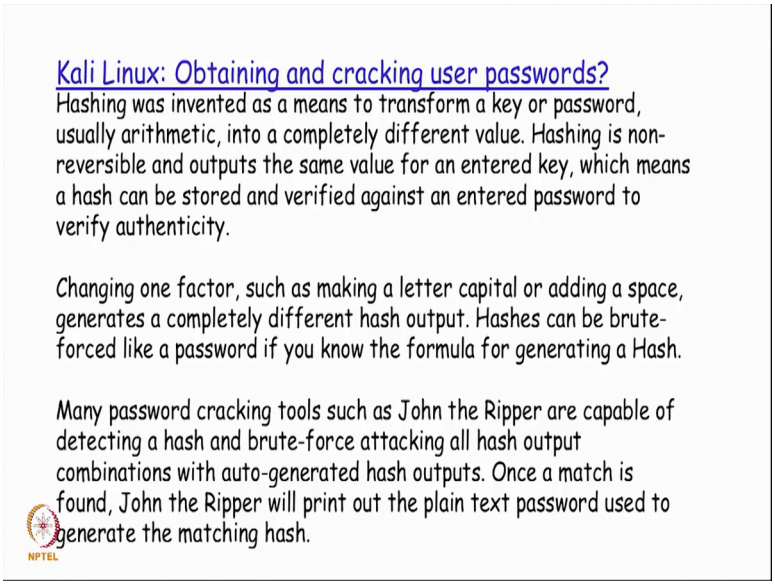


Passwords are typically expected to be stored on the system so that a very the users identity and access rights could be verified, right. Now, as compared to the practices that was actually existing a few decades back it is no longer the passwords are no longer actually stored as in plain text files today, right. So, and again you although there is a general tendency that the passwords are going to be stored in a file and that file is encrypted, it is not a practice generally followed today because of the challenges associated with storing and then using the key that is required for decrypting that text file in which the passwords are containing rights.

So, that basically becomes a key distribution issue and also possibly expose a weakness to the potential attacker by if for example, the attacker could basically try to get the key through some source then and that particular system is actually having let us say hundred users it is given then that all the hundred users authentication is basically going to be now getting compromised, right, with the single key determination by the attacker which

is basically what has been used to do the encryption. If that is not been done today of actually storing the password. So, what is a method that is generally adopted is basically what is called as hashing. So, some of us would have actually read about and also worked with a different kind of hashing algorithms or heard about it saying like MD5 is an hashing algorithm SHA-256 and so on, right.

(Refer Slide Time: 06:29)



Now, what exactly is hashing? Now, basically hashing is a non-reversible output that is actually got for a given input. Now, why is it that the hash is actually being stored here is that the when the password is actually entered by the user the string that has been entered as a password the hash value will be computed on that and since a hash is actually stored right now they will now be compared to see if they match. Now, if these two match it is it is expected that the password that has been actually given by the user right now is the correct one and then authentication is provided to that particular user, right.

Now, how is that sort of deterministic is that the basic mechanism by which the hashing algorithm is implemented is that algorithm is actually implemented in such a manner that even if I actually change one character in my password, right or for that matter even a few of those bits of that character the hash value that the algorithm is going to generate finally for me it is not going to be actually the same, right.

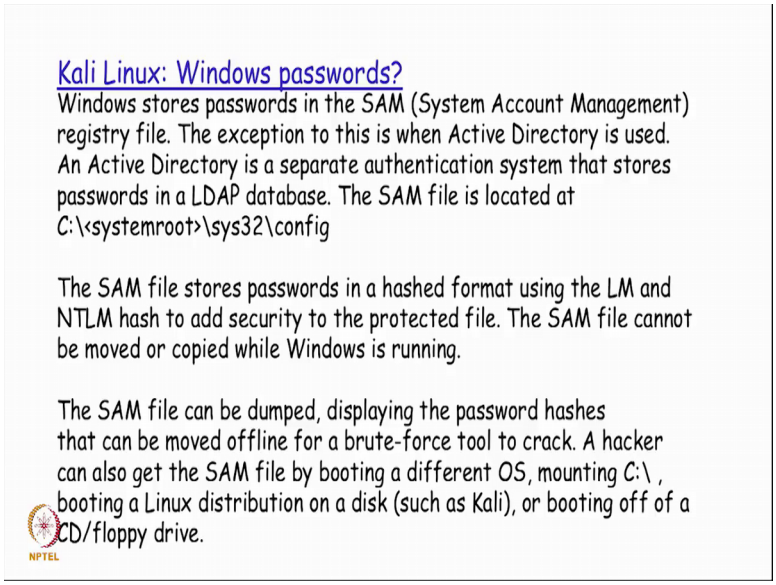
So, in that respect, if I basically try to have a string let us say with the first character as capital letter and I the user is basically now typing the password is the same string, but

this user is not right now typing the first character of that password string in capitals, but is typing it in small letters the hash value that is going to be generated for this typed password is not going to be matching with the hash value that is stored for that same string with the capital letter with a with a with the first letter starting in capital letters, right. So, to that extend the algorithms are implemented in such a manner and design in such a manner that even a single byte or a few bits of the byte is actually changes then the hash value the that is going to be actually getting generated now for the given input string is not going to be matching with that of the stored password.

So, thereby it is actually sort of proved that it is a non-reversible mechanism meaning to say that with the hash value it is not possible to re-generate back the original string and secondly, the hash value itself is going to change even if a few bits of my string actually changes, right.

So, that is basically the beauty of this algorithm and the most of the cracking tools that for example, the tool of John the Ripper that we have seen also basically uses this technique and principle that ones I determine the hash value of the password that has been stored and the hash value of the string that has been entered right now to be the same then it can be safely assumed that the tool has successfully cracked the password for that particular user. So, that is basically the underlying mechanism by which the tool is actually working.

(Refer Slide Time: 09:33)



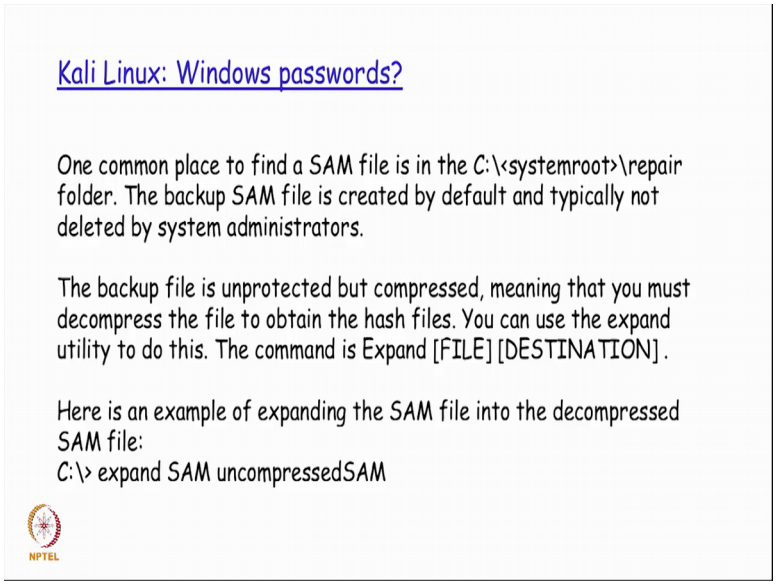
Now, coming to the passwords on how it is actually stored and validated on windows operating system. So, windows basically stores passwords in a file called as a SAM though Systems Account Management registry file and this password is actually stored if the windows is right now trying to actually use local users created on that particular windows system. On the other hand if I basically have a centralized user authentication mechanism like an active directory kind of a service then this entire thing is not actually applicable because the whole user password combination and a whole set of authentication mechanisms are actually stored in a centralized manner in a different format whenever we use active directory.

So, the active directory is basically a customized version of the open source directory based authentication protocol called LDAP. So, windows Microsoft windows has basically customize the LDAP and then named it is an active directory by which the all the users in a network typically would be actually authenticated by a single server running as a active directory server on the on the network and there will be no local user name password combinations in that particular kind of a setup. So, the SAM file if it is actually if it is basically the windows system using the local user authentication only the SAM file is basically stored stores the passwords in a hash format and the algorithms hash algorithm that are typically used in windows is LM or NTLM hash algorithm, right.

So, then the windows OS is running one of the important restrictions that it actually imposes is that since it is actually going to be locking the SAM file because it is going to be accessing it on a continuous basis this particular registry file will not be allowed to be moved or copied at the time the windows OS is running, right. So, this is basically becomes a limitation from the point of view of doing a penetration testing.

So, in order to overcome this what is actually done is that if it all possible that particular windows system is multi bootable with another operating system like Linux the system could be booted onto the Linux OS after which the file that is actually use a SAM registry file that is actually used for a authentication could be very easily read because windows is right now not running and locking that file, right. So, that is one method by which a typically you could get access to the contents of the SAM file right.

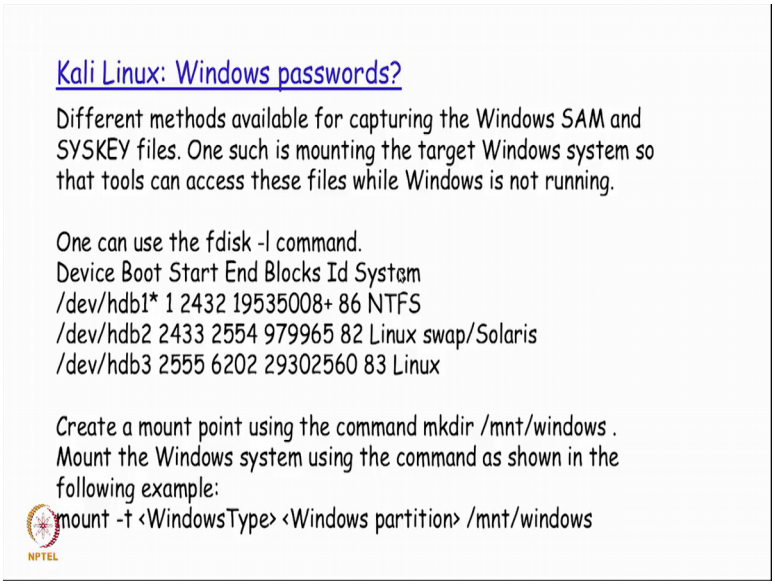
(Refer Slide Time: 12:19)



So, once a SAM file is accessed the backup file or the SAM file is sort of in an unprotected form, but compressed. So, it has just got to be decompressed which is actually very very easy to do so. For example, one of the commands that you can actually run as the command called expand which will basically do a sort of a decompression of the given file and then copy it into another file name that is actually specified there.

So, in this decompressed file that is given as the second argument to the expand command I will basically have the contents of the original SAM file in all in all easily accessible as a plain texter form. Now, another method that I could potentially make use of is by trying to determine from the other operating system what are the different disk partitions actually available right.

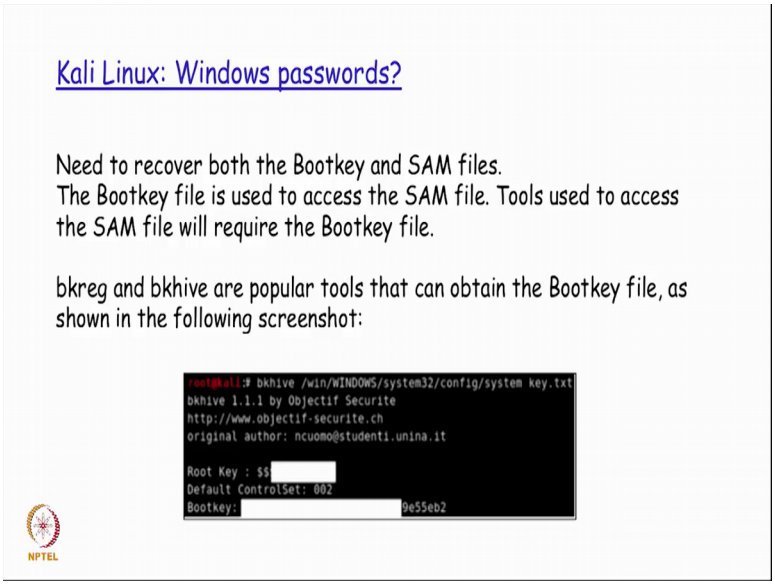
(Refer Slide Time: 13:12)



So, when I actually try to find out the disk partitions that are available using the f disk minus l command on Linux for example, it basically list down the disk partitions hdb 1, hdb 2, hdb 3 for example, here and then says our each of those what is a file system or the OS that is possible recognizing them, right. So, in this case you find that NTFS is mentioned for the first disk partition which means that this NTFS is basically a file system which is typically supported by windows system and this should be the partition which should actually the mounted now to ensure that we now have a hook to get the access of the of the of the particular SAM registry file.

So, you just mount that particular windows partitions on this particular example it is going to be slash dev slash hdb 1 on to a mount point call slash mnt windows, right and along with that you also specify the windows the type that is basically the file system type with a minus t option to the mount command, right. So, once we do this successfully as a super user on Linux for example, we will now have access to the system registry file, right.

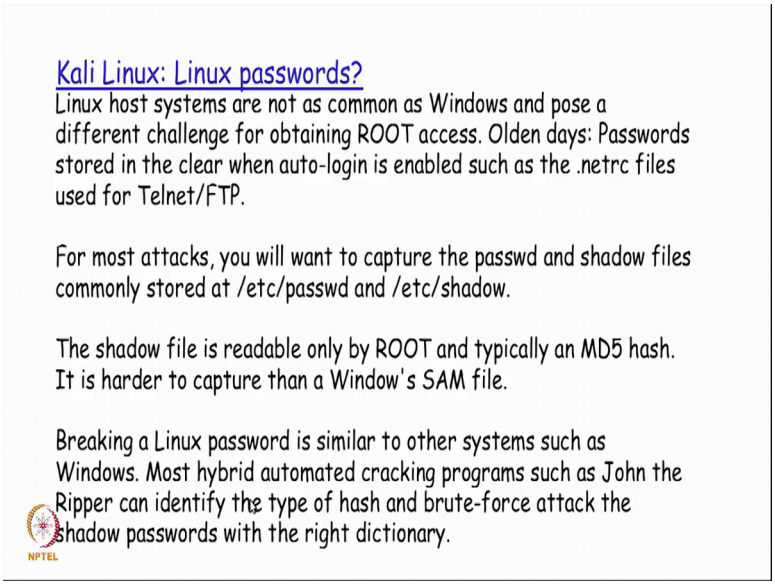
(Refer Slide Time: 14:19)



Now, the later versions of windows basically also uses what is called as a boot key file because a boot key file is actually used to access the SAM file, right. So, now, we need to access first get access to the boot key after that only will be able to access the SAM file because for accessing the SAM file you will need the contents of the boot key. So, now, on the windows how do you actually access get access to the contents of the boot key file is there is a command call bkreg and bkhive. So, if you have a screenshot here so, if you basically say that bkhive and give the config system where the other windows is actually using it.

So for example, in our case if we are mounted it on win windows and system thirty twos is basically the default directory of the system folder within windows. So, you just give this command and after which it will basically display you the contains of the root key and the boot key. So, here it is actually been grade out for it not to be getting displayed out right, but on a on a on a properly running a system where you could successfully run the bkhive command you will have the output of the boot key and also the entire boot key string format also displayed as part of the command output, right.

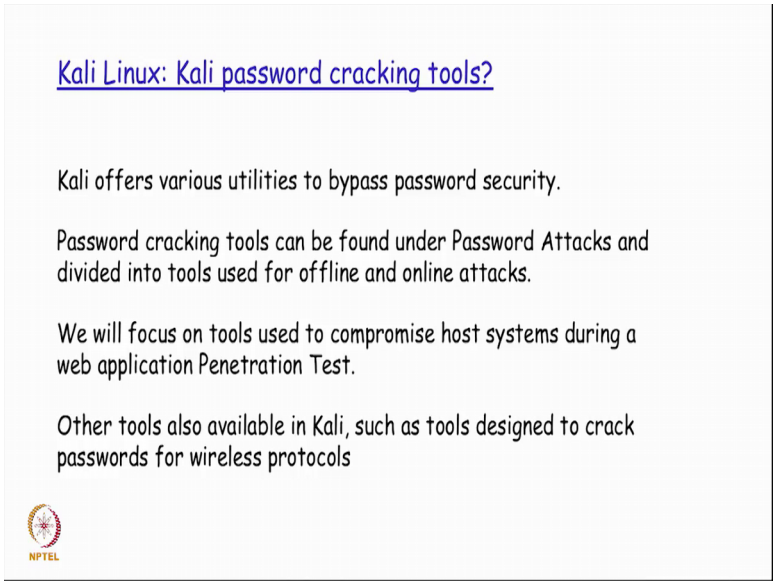
(Refer Slide Time: 15:42)



Now, coming down to the Linux passwords so, with the Linux host system you will typically need to have the root access because the files of the password in the shadow that is actually used., for example, the shadow file is actually readable only by root and also typically uses a MD5 hash, right. So, it is little bit more difficult to actually crack as compared to a windows SAM file right.

Now, we will see subsequently in the session down below on how we could actually use GOY interface with the John the Ripper tool that will sort of smoothen out our cracking of a Linux password mechanism right. So, there are different ways by which I could make use of it and. So, we will actually take a look at that, right.

(Refer Slide Time: 16:33)



So, Kali basically password actually offers a different kind of password cracking tools. So, we are going to see a few of the tools and in the kali Linux distribution you will typically find that available under the password attacks and within that you have two classifications one set of tools that could be used for offline and another set of tools that could be used for online attacks, right. So, depending on the target that we are trying to run this password cracking tools on and the environment with in which the target is operating the attacker would possibly decide which out of the tools he will basically try to run to crack the password.

So, it could be either tool in the offline category tools or it could be a tool in the online category tools right we will see subsequently of the tools that are typically used for doing a web application penetration testing and there are also other tools available in Kali Linux which is also designed to crack passwords for Wi-Fi protocol. So, for example, if you are basically having a laptop device which with which you are trying to connect to a Wi-Fi access point as we all know we need to give the key for connecting successfully into the Wi-Fi access point, right.

So, there are tools that are specifically targeted for trying to determine those to those keys to crack them and make use of it subsequently by the attacker. So, there are set of tools which are actually designed for that purpose as well.

So, in sub sequent session we will actually be seeing some of the tools as an example as we have been iterating before it is very much required for you to actually try out the different kind of tools and the various options that are possible in that in order to make yourself comfortable with using these tools because a reading through a set of presentation slides or reading through a documentation a man pages very easy as compared to actually trying the tool itself and experimenting with that because that is basically what is actually going to be making our fundamentals much more stronger.

So, we will actually re-iterate our previous point across to you where in you need to really practice the tools that are being talked about as well as the rest of the tools it are available in the same category. So, that you really get a hands on experience and get more comfortable in basically how the tool is working as well as how and what kind of an environment that tool will actually be advantageous to meet our objectives.

Thank you.