

## Class Four

### Topics:

- Additional Identification & Authentication Methods
- More Complex Authentication Schemes
- Authentication Products
- Access Control
- Discretionary vs. Mandatory Access Control
- Ownership & Custody
- ACLs in Common Systems
- Creating a Workable Access Control System
- Extra: ACLs in Windows XP
- Extra : Setting up one-time passwords in Linux with OPIE

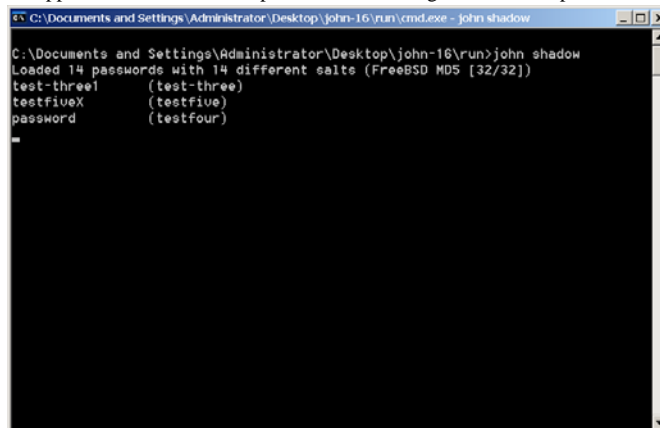
8/24/2007

Class 4

1

## Example of a password dictionary attack

Output from John The Ripper from: <http://www.openwall.com>  
Applied to a Unix shadow password file having several weak passwords



```
C:\Documents and Settings\Administrator\Desktop\john-16\run\cmd.exe - john shadow
C:\Documents and Settings\Administrator\Desktop\john-16\run>john shadow
Loaded 14 passwords with 14 different salts (FreeBSD MD5 [32/32])
test-three1 (test-three)
testfiveX (testfive)
password (testfour)
```

8/24/2007

Class 4

2

## Additional Forms of Authentication

### “Something you know”

- Associative Response

### “Something you have”

- Card Key
- One Time Passwords (using some form of cryptography, generated by a device you possess)
  - True One-Time Passwords
  - S/Key
  - Synchronous (Time-based)
  - Asynchronous (Challenge-Response)
- Proximity Detectors

### “Something you are”

- Biometric
  - Retina Scan
  - Fingerprint
  - etc

**Two Factor Authentication** - combines two separate factors of authentication for more reliability

8/24/2007

Class 4

3

## Additional Forms of Authentication

- Two factor authentication:
  - Combines two factors of authentication into a single mechanism
  - Attempts to overcome weaknesses of a single method:
    - “Some you have” is often “Something to lose or steal”
    - “Something you know” is often “Something easy for someone else to guess”
  - Examples:
    - Token that requires a password
    - Biometric plus passcodes

8/24/2007

Class 4

4

## Biometrics

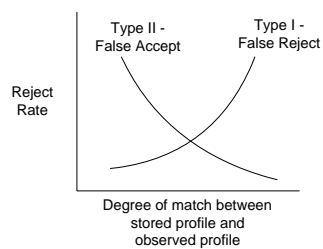
8/24/2007

Class 4

5

## Issues with Biometrics

- False reject of valid individuals (Type I Error or "insult" rate)
- False accept of imposters (Type II error or "imposter" rate)
- Zero Effort False Accept - Likelihood of authenticating as another individual without any deliberate effort to commit forgery of biometric



Note that:

There is an inevitable trade-off between Type I and Type II error rates

Neither rate becomes zero, under practical circumstances

The degree of "tuning" (optimal mix of Type I and Type II) depends on site security requirements

8/24/2007

Class 4

6

## Issues with Biometrics

- Some portions of the population may be unable to use a specific biometric
- User acceptability (a major issue)
- Extra cost (becoming less of an issue)
- Integration with existing authentication mechanisms
- Difficult, if not impossible, to change authenticator if stolen by imposter (replay attacks)
- User may provide authentication without knowing (potential for fraud in false authorization of financial transactions)
- Reliable enough to use as legal evidence?

8/24/2007

Class 4

7

## One-Time Passwords:

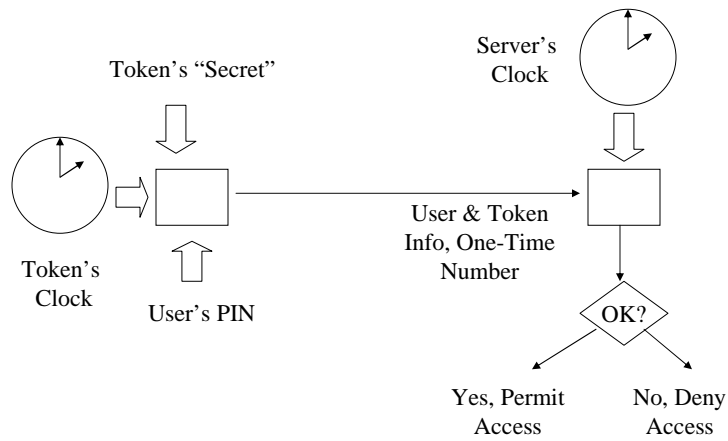
- True One-Time Pad
  - Unbreakable in theory and in practice
  - Unwieldy
- S/Key:
  - Mimics one-time pad through repeated application of one-way hash to a "seed"
  - The  $n+1$  hash is derived from the  $n$  hash, but the reverse cannot be done
  - You use the  $n$  hash first, then the  $n-1$ , then  $n-2$ , etc. The number  $n$  (the sequence number) is exchanged as part of the protocol. Once a given  $n$  is used, it is never re-used. When you run out of  $n$  (when  $n$  is finally decremented to zero), you need to start over by re-generating the sequence.
  - Hash is based on a shared secret, usually a password
- S/Key is similar in use to "99 Bottles of Beer on the Wall"
- S/Key is a standard described in IETF RFC 1938

8/24/2007

Class 4

8

## How Synchronous Authentication Works "Based on the time"

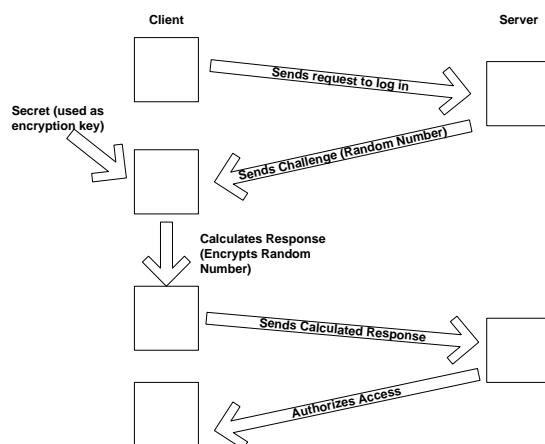


8/24/2007

Class 4

9

## How Challenge-Response Works (Asynchronous) "Based on a shared secret"



8/24/2007

Class 4

10

## Proximity detectors

- Carried with the user at all times
- Small radio transmitter that client workstation senses
- When the user walks away, the workstation is automatically locked

Product references:

- XyLoc from Ensure Technologies
- Bio Proximity Security System from Access Denied Systems

8/24/2007

Class 4

11

## Proximity Detectors (example)



8/24/2007

Class 4

12

## More Complex Authentication Schemes:

- How to authenticate to a network when there are many possible points of entry?
- How to manage very large numbers of users attempting to access a network?
- How to accurately track usage for billing purposes (very important for old-style online access vendors)?
- How to allow single authentication for access to a variety of services, offered by different servers?
- How to guard against eavesdropping and other attacks against network authentication?
- How to facilitate managing the identities of large numbers of users in very large enterprise networks?

8/24/2007

Class 4

13

## More Complex Authentication Schemes:

- Single Sign-on
- Kerberos
- Remote Authentication Protocols
  - RADIUS
  - TACACS+, XTACACS

8/24/2007

Class 4

14

## Single Sign On Product Types

- Tokens or Credentials (including Kerberos and PKI-based solutions)
- Workstation Logon script
- Authentication Server Scripts
- Issues:
  - Password Storage
  - Target System Log-On
  - Single point of failure
- Tendency:
  - Token based
  - Uses enterprise directories to store credentials

8/24/2007

Class 4

15

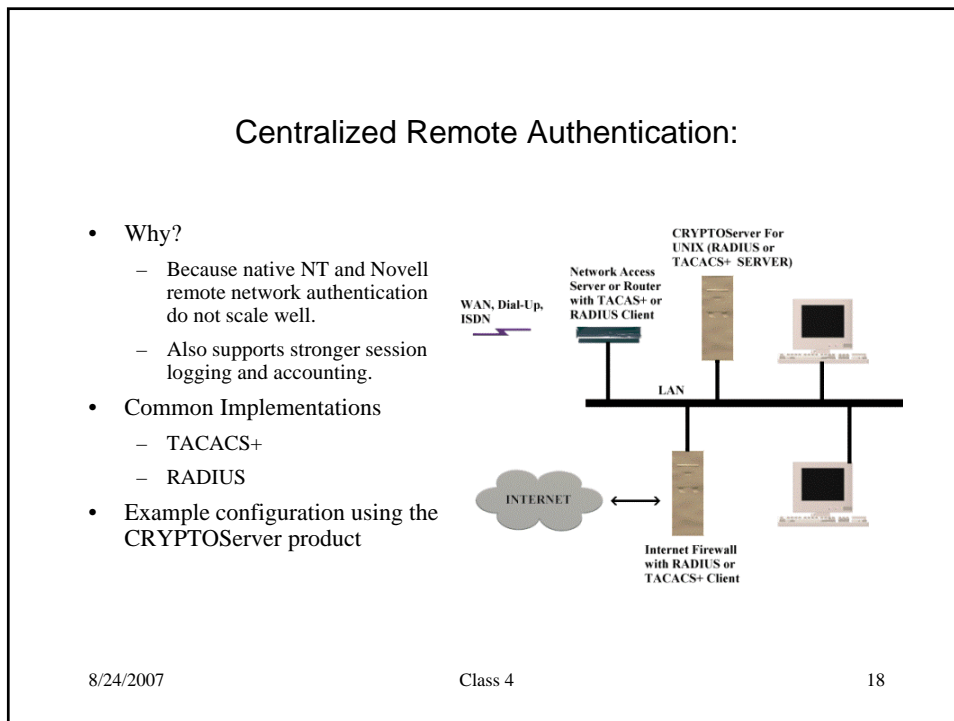
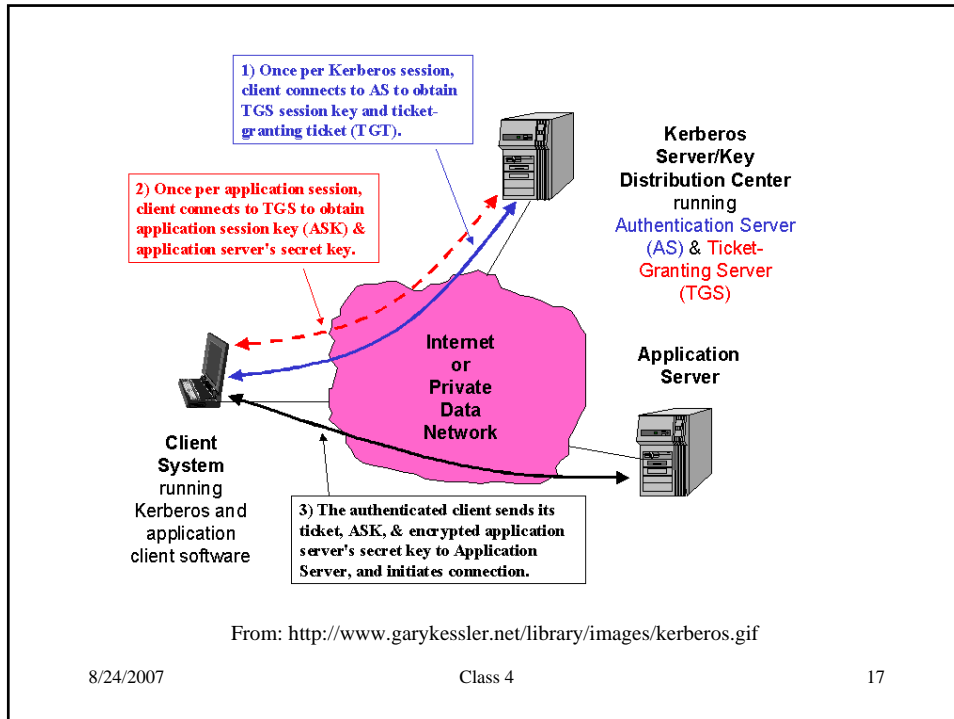
## Kerberos

- Originally design at MIT as part of Project Athena
- Uses a centralized Key Distribution Center
- Session keys are generated at sign-on, for a limited time period
  - The user's password is used to generate their master key, which is used to request a session key
- At session start, a ticket-granting ticket (TGT) is provided. This contains the session key and expiration time
- The TGT is then used to request services from other servers. The client must request a ticket for that specific service. Once obtained, the ticket permits use of the service for the specified time.
- Kerberos uses symmetric key encryption. The Key Distribution Server stores all encryption keys. This server must be secured.

8/24/2007

Class 4

16



## How does RADIUS Work

- Acts as “manager” to other remote access servers (RAS servers)
- RAS server forwards authentication request to RADIUS server
- RADIUS server approves authentication
- RAS server then accumulates session accounting information
- When session ends, RAS forwards accounting information to RADIUS server
- Multiple RADIUS servers may forward authentication requests to each other via proxy authentication
- RAS server to RADIUS server traffic is encrypted

8/24/2007

Class 4

19

## Access Control

- Now that we know who you are, how do we manage what you are allowed to do?
  - Networks
  - Servers
  - Server operations and management commands
  - Applications
  - Application functions

8/24/2007

Class 4

20

## Access Control for Data

- Permit or deny access to data, or any system resource
- Types of data access:
  - Read, write, delete entire file
  - Execute
  - Create, copy, move
  - Update, add, delete contents
  - Test existence
  - Modify attributes (e.g., last date modified, permissions, etc.)

8/24/2007

Class 4

21

## Discretionary vs. Mandatory Access Controls *Access Control Policies*

- Discretionary Access Control (DAC)
  - Creator or owner of file ultimately decides security
  - Owner may "give away" file if so desired
- Mandatory Access Control (MAC)
  - Users and files have labels, assigned by the system
  - System-wide policies determine certain access based on labels
  - Owner still has rights to files they create, though more circumscribed than with DAC
  - Owners may (sometimes) be able to change the sensitivity label of files they create
  - MAC prevents tricking owners into giving away their files in violation of security policies
  - Based on notions of clearance & classification, which must be used by the operating system for determining access rights

8/24/2007

Class 4

22

## Ownership and Custody

- Owner - Individual responsible for (accountable for) data protections
  - Statutory authority responsible for data (DoD 5200.28)
  - Entity responsible for data, which must communicate security-relevant needs to users and custodians of the data
- Custodian - Individual entrusted with the day-to-day security of the data
  - May change frequently as data undergoes various processing steps

8/24/2007

Class 4

23

## Ownership and Custody

- Per Roger Shaw's article (in lecture notes):
  - Owner - The person responsible for making and communicating decisions regarding the use, classification, and protection of assigned information. The owner has property rights interest in an information asset for an organization.
  - Custodian - The person who has authorized possession of the information and is responsible for owner-authorized controls. This is often Information Services.
- Per Fites
  - Owner - ... has responsibility for specific data types and is charged with communication of the need for certain security-related handling procedures to both the users and custodians of the data.
  - Custodian - ... has been entrusted with the possession of, and the responsibility for, the security of specified data.

8/24/2007

Class 4

24

## Privileges vs. Rights

- Right – capability to access a specific object
- Privilege – ability to execute commands or functions
- Privileges may be pre-defined in the operating system. They are often used to define system management roles (e.g., operator, network engineer, system administrator, etc.)

8/24/2007

Class 4

25

## The Rule of Least Privilege

- Each subject has the most restrictive (most minimal) set of privileges necessary to get the job done.
- Limits damage caused by accident, error, or intentional action.
- Consider applications of least privilege to:
  - Subjects which are processes, or software in execution
  - Subjects which are human beings
- “Timely Revocation of Trust”
  - Least privilege defined in time. Privileges are only granted when needed, and are revoked when task is complete.

8/24/2007

Class 4

26

## Granting Authority Within a Program or Process

- Temporarily allows a user higher authority to perform a specific task
- Ideally done without providing permanent or uncontrolled access
- Example:
  - A systems operator should not be able to read sensitive files, but requires temporary read authority while executing a backup routine
- Often implemented by giving the user of a program the rights of the program owner, only while executing the program
- Feature provided in many operating systems:
  - OS/400 - "adopting of authority"
  - UNIX - SUID and SGID
  - Windows 2000/XP – "runas" command

8/24/2007

Class 4

27

## Methods for Implementing Access Control (inside the Operating System)

- Resource Passwords
  - To access a resource requires a password
  - Each resource has its own password, which is shared by all users requiring access
  - Difficult to administer and use
  - Little used - mainly historical value
- Capabilities
  - Like a key ring possessed by a user. A user may "loan" or pass their capabilities to another user
  - Used internally in the IBM S/38
- Access Control Lists
  - A list of users and privileges associated with an object
- Permission Bits
  - String of bits, defining access permissions for pre-defined and fixed groups of users
  - "Degenerate" case equals ACLS

8/24/2007

Class 4

28

## The Access Control List (ACL)

- Attached to the data or other object
- Defines access rights by:
  - Group
  - Individual user
  - Public/World/Anyone Not Otherwise Defined
- Sequential comparison is made, first match results in user being granted or denied access

8/24/2007

Class 4

29

## The Access Control List compared to Capabilities

- ACL
  - Will specify, for a given object, the access rights of each subject on the system.
  - Associates list of subjects with a specific object.
  - are part of the object's definition
  - correspond to the columns in an access matrix
- Capability
  - For a given subject, gives all the access rights to various objects.
  - Associates lists of objects with a specific subject
  - Correspond to the rows in an access matrix.
- Access Matrix – Shorthand for expression users' access to objects, where each user is given a row, and each object a column, and the cell contains the access right.

8/24/2007

Class 4

30

## The Access Control List compared to Capabilities

- Capability - An unforgeable ticket that is accepted by the system as proof that the presenter has the right to access the object specified in the ticket. It is often interpreted by the operating system and the hardware as an address for the object. The ticket contains information describing the nature of the permitted access (e.g., read-only, etc.). **Possessed by the User**
- Access Control List - A list of entities (subjects) and their access rights to a specific object. **Attached to the Object.**

8/24/2007

Class 4

31

## The Access Matrix

- Requires categorization of subjects and objects
- Object categorization is by similar use, common use in supporting a process
- Subject categorization is by roles, what tools you require to accomplish your job
- Sample:

Subjects	Objects			
	Production Data	Production Programs	Development Programs	Operating System Programs
End User	RW	X	-----	X
Programmer	R	R	RWX	X
Operations	R	RW	R	RX
System Programmer	R	R	R	RWX

**Note: In this example, Operations performs backup and production change management. Also note that columns represent ACLs, while rows would represent Capabilities.**

8/24/2007

Class 4

32

## How to Create a Working Access Control System

- Unique user identification
- Group similar users
- Naming conventions for all resources
- Classify data (by the nature of use)
- Restrict any system utilities which may permit circumventing ACL mechanisms
- No exceptions (not even vendors & system programmers)
- Production change control, separation of development and production

8/24/2007

Class 4

33

## **Extra Material:** How ACLs are Implemented in Common Systems

- MVS/RACF
  - Called “access authority”
  - Different types for dataset (file) or general resources
  - Resources have a “universal access authority”, similar to UNIX World rights
- AS/400
  - Called “object authority”
  - Separately supports both data operations (read, write, update) and object management operations (copy, create, delete)
  - Named combinations facilitate assigning access

8/24/2007

Class 4

34

## Extra Material: How ACLs are Implemented in Common Systems

- UNIX
  - Natively supports only permissions (Read/Write/Execute for Owner/Group/World)
  - Some versions support fully functional access control lists

8/24/2007

Class 4

35

## Extra Material: How ACLs are Implemented in Common Systems

- Windows NT/2000/XP
  - Read, Write, execute, Delete, Change Permission, Change Owner
  - Directory ACLs propagate to subdirectories, and apply to newly created files
  - Named combinations facilitate assigning access
  - “Everyone” user pool is inclusive

8/24/2007

Class 4

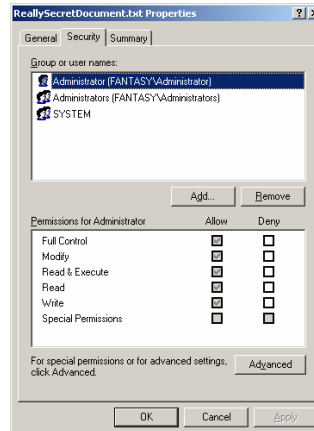
36

## Extra Material: A look at Windows XP ACLs

For this file:

- Owned by the Administrator
- The Administrator is individually granted access
- Two groups (Administrators and SYSTEM) are also granted rights

Clicking *Advanced* will tell us more...



8/24/2007

Class 4

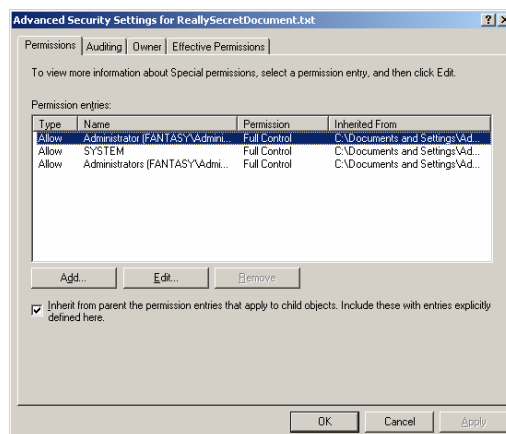
37

## Extra Material: A look at Windows XP ACLs

*Advanced* shows:

- Type (allow or deny)
- Name of user or group
- Permission
- Where permission is inherited from
- And a checkbox that allows the object to inherit permissions from the parent object (the folders above it)
- So you can make a lot of objects have the same access by placing them in the same folder and setting the folder's ACL

- Press *Edit*



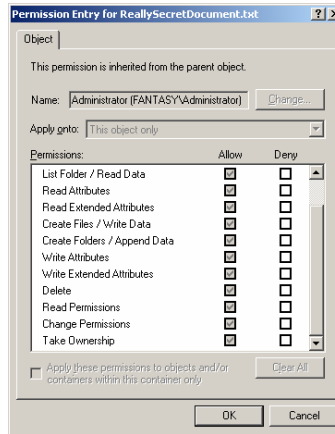
8/24/2007

Class 4

38

## Extra Material: A look at Windows XP ACLs

*Edit* shows detailed permissions of a single user or group.



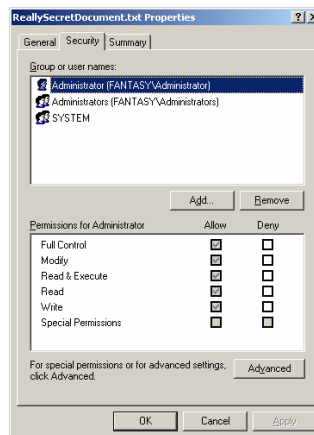
8/24/2007

Class 4

39

## Extra Material: A look at Windows XP ACLs

A new text file with default ACLs Created by the Administrator



8/24/2007

Class 4

40

## Extra Material: One time passwords under Fedora Core 5 Linux

Thanks to <http://www.waldner.priv.at/opie.html>  
for working instructions

8/24/2007

Class 4

41

## Purpose

- Replace reusable passwords for Linux user authentication with one-time passwords
- Use free open source resources
- Demonstrate how to configure alternative authentication methods using PAM
- Demonstrate the S/Key and Opie password generation methods

8/24/2007

Class 4

42

## What is PAM?

- Pluggable Authentication Modules
- A UNIX standard for adding and managing 3<sup>rd</sup> party authentication modules
- Allows using different authentication methods for different services
- Provides the “glue” between the service (ssh, telnet, su, etc.) and the authentication method (passwords, tokens, central RADIUS server, etc.)

8/24/2007

Class 4

43

## What is Opie?

- Onetime Passwords In Everything
- Based on S/Key
  - Repetitive application of a one-way hash to an initial shared secret
  - Use in reverse order to prevent malicious parties from predicting the next password
- Developed at and for the United States Naval Research Laboratory (NRL)
- Couldn't be called “S/Key” due to trademark issues

8/24/2007

Class 4

44

## Needed

- Latest Fedora Core installation.
- Opie (One Time Passwords in Everything) rpm module:
  - opie-2.4-551.i586.rpm from <http://rpmfind.net/linux/RPM/suse/9.3/i386/suse/i586/opie-2.4-551.i586.html>
- To install Opie:  

```
rpm -iv opie-2.4-551.i586.rpm
```

8/24/2007

Class 4

45

## What's in Opie?

- pam\_opie.so – a Pluggable Authentication Module (PAM) to allow Opie to easily govern different authentication processes
- opiepasswd – Defines a user to Opie, and allows resetting the secret passphrase
- opieinfo – Gives the seed and sequence number for a user
- opiekey – Opie passphrase calculator. Opiekey will generate a one time password (OTP) given a sequence number, seed, and passphrase as input
- opielogin, opiesu, opieftp – Replacements for login, su, and ftp that are “opie-ized”. We will use PAM instead.

8/24/2007

Class 4

46

## Set up a user

- Add the user to Linux:  
`/usr/sbin/useradd opieuser`
- Define the user to Opie:  
`/usr/bin/opiepasswd -c opieuser`
  - Enter a secret passphrase. This phrase will be required for future login attempts
  - When done Opie will display the initial sequence number, seed, and OTP for your first login

8/24/2007

Class 4

47

## Configure PAM

- Edit the file `/etc/pam.d/ssh` :
- Take out `common-auth`
- Insert `auth sufficient pam_opie.so`
- Insert `auth required pam_deny.so`

8/24/2007

Class 4

48

## Configure SSH

- Edit `/etc/ssh/sshd_config` :
  - set the following:  
`ChallengeResponseAuthentication yes`  
`PasswordAuthentication no`  
`PAMAuthenticationViaKbdInt yes`  
`UsePAM yes`  
`UsePrivilegeSeparation yes`
- Restart sshd :  
`/sbin/service sshd restart`

8/24/2007

Class 4

49

## To Login using Opie

- Open two terminal windows on the Fedora Core system
- In one window, try to login using ssh :  
`ssh opieuser@localhost`
- You will get a prompt giving you the seed and sequence number for your next OTP:  
`otp-md5 496 cr4535`  
Response:
- The seed is 'cr4535', the sequence number is "496"

8/24/2007

Class 4

50

## To Login using Opie

- In the other window, use the opie calculator opiekey to figure out what your OTP should be:  
    `opiekey 496 cr4535`  
    Using the MD5 algorithm to compute response.  
    Reminder: Don't use opiekey from telnet or dial-in sessions.  
    Enter secret pass phrase:  
    `IF NOVA REEF AUTO IFFY SONG`
- Enter the sequence number and seed on the command line. Enter the pass phrase when prompted (it will not display). Opie will provide a set of English words. These are your OTP (in this case “IF NOVA REEF AUTO IFFY SONG”)

8/24/2007

Class 4

51

## To Login using Opie

- Type your OTP (in this case “IF NOVA REEF AUTO IFFY SONG”) as your response to the login prompt.
- If you did everything right, you will be logged in.

8/24/2007

Class 4

52