Tedu.cn
达内教育

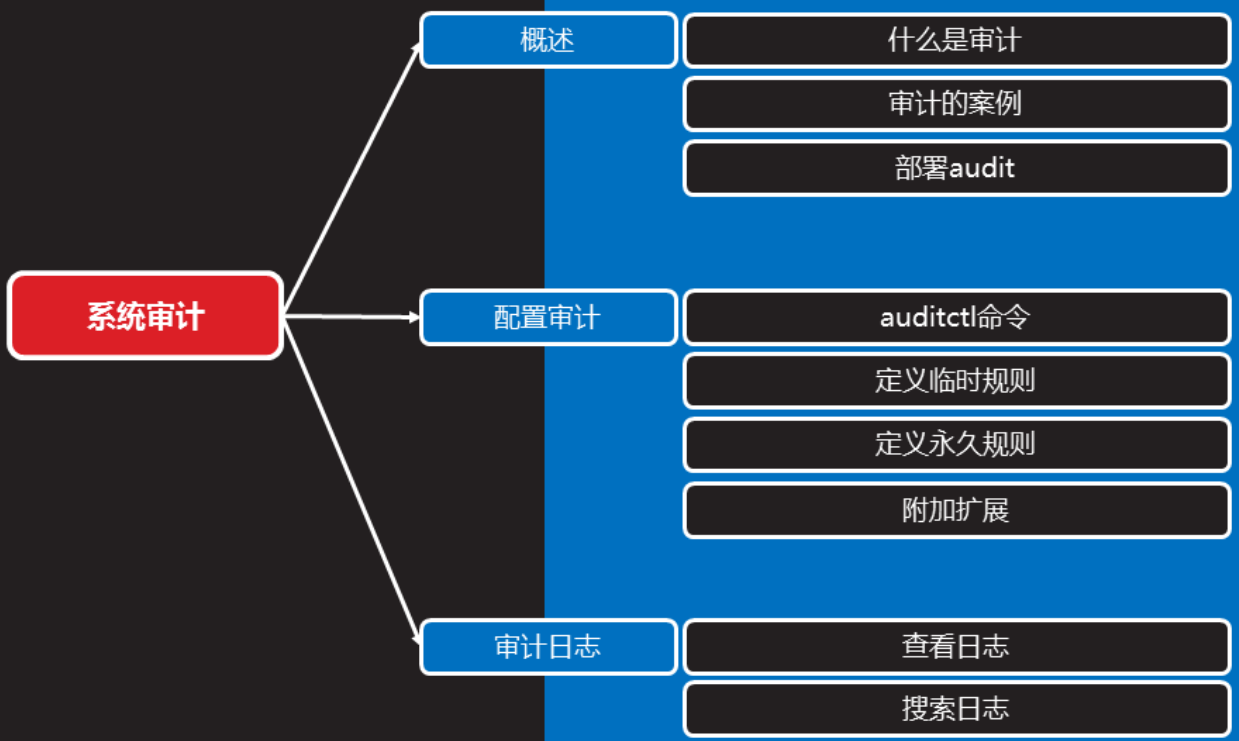
服务安全与监控

NSD SECURITY DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	系统审计
	10:30 ~ 11:20	
	11:30 ~ 12:00	服务安全
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Linux安全之打补丁
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑

系统审计



Tedu.cn 达内教育

概述

Tedu.cn 达内教育

什么是审计

知识讲解

- 基于事先配置的规则生成日志，记录可能发生在系统上的事件
- 审计不会为系统提供额外的安全保护，但她会发现并记录违反安全策略的人及其对应的行为
- 审计能够记录的日志内容：
  - 日期与事件、事件结果
  - 触发事件的用户
  - 所有认证机制的使用都可以被记录，如ssh等
  - 对关键数据文件的修改行为等

审计的案例

知识讲解

- 监控文件访问
- 监控系统调用
- 记录用户运行的命令
- 审计可以监控网络访问行为
- ausearch工具，可以根据条件过滤审计日志
- aureport工具，可以生成审计报告

部署audit

知识讲解

- 使用审计系统需要安装audit软件包
- 主配置文件为/etc/audit/auditd.conf

```
[root@svr7 ~]# yum -y  install  audit
[root@svr7 ~]# cat /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log                    //日志文件
[root@svr7 ~]# systemctl start auditd
[root@svr7 ~]# systemctl enable auditd
```

Tedu.cn 达内教育

配置审计

Tedu.cn 达内教育

auditctl命令

- auditctl命令控制审计系统并设置规则决定哪些行为会被记录日志

知识讲解

```
[root@svr7 ~]# auditctl  -s          //查询状态
[root@svr7 ~]# auditctl  -l          //查看规则
[root@svr7 ~]# auditctl  -D          //删除所有规则
```

定义临时规则

知识讲解

- 定义文件系统规则，语法如下：
  - auditctl -w path -p permission -k key_name
  - path为需要审计的文件或目录
  - 权限可以是r,w,x,a(文件或目录的属性发生变化)
  - Key_name为可选项，方便识别哪些规则生成特定的日志项

```
[root@svr7 ~]# audit -w /etc/passwd -p wa -k passwd_change
//设置规则所有对passwd文件的写、属性修改操作都会被记录审计日志
[root@svr7 ~]# audit -w /etc/selinux/ -p wa -k selinux_change
//设置规则，监控/etc/selinux目录
[root@svr7 ~]# audit -w /usr/sbin/fdisk -p x -k disk_partition
//设置规则，监控fdisk程序
```

定义永久规则

知识讲解

- 定义永久规则
- 写入配置文件/etc/audit/rules.d/audit.rules

```
[root@svr7 ~]# vim /etc/audit/rules.d/audit.rules
-w /etc/passwd -p wa -k passwd_changes
-w /usr/sbin/fdisk -p x -k partition_disks
```

附加扩展

知识讲解

- 扩展知识
  - ：通过审核也可以监控系统调用

```
[roo@svr7 ~]# cat /usr/include/asm/unistd_64.h
```

  - 规则参考模板

```
[root@svr7 ~]# ls  /usr/share/doc/audit-版本号/rules/
```

审计日志

查看日志

知识讲解

- 定义规则

```
[roo@svr7 ~]# auditctl -w /etc/ssh/sshd_config \
>-p warx  -k  sshd_config
```

- 查看日志

```
[roo@svr7 ~]# tailf  /var/log/audit/audit.log
type=SYSCALL msg=audit(1517557590.644:229228):  arch=c000003e  syscall=2
success=yes exit=3 a0=7fff71721839  a1=0 a2=1fffffffffff0000  a3=7fff717204c0
items=1 ppid=7654  pid=7808  auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
egid=0 sgid=0 fsgid=0 tty=pts2 ses=3 comm="cat" exe="/usr/bin/cat"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  key="sshd_config"
```

查看日志（续1）

知识讲解

- type为类型
- msg为(time_stamp:ID)，时间是date +%s
- arch=c000003e，代表x86_64（16进制）
- success=yes/no，事件是否成功
- a0-a3是程序调用时前4个参数，16进制编码了
- ppid父进程ID，如bash，pid进程ID，如cat命令
- auid是审核用户的id，su - test, 依然可以追踪su前的账户
- uid，gid用户与组
- tty:从哪个终端执行的命令

查看日志（续2）

知识讲解

- comm="cat"　　用户在命令行执行的指令
- exe="/bin/cat"　　实际程序的路径
- key="sshd_config"　　管理员定义的策略关键字key
- type=CWD　　用来记录当前工作目录
  - cwd="/home/username"
- type=PATH
  - ouid(owner's user id)　　对象所有者id
  - guid(owner's groupid)　　对象所有者id

搜索日志

知识讲解

- 系统提供的ausearch命令可以方便的搜索特定日志
  - 默认该程序会搜索/var/log/audit/audit.log
  - ausearch options -if file_name可以指定文件名

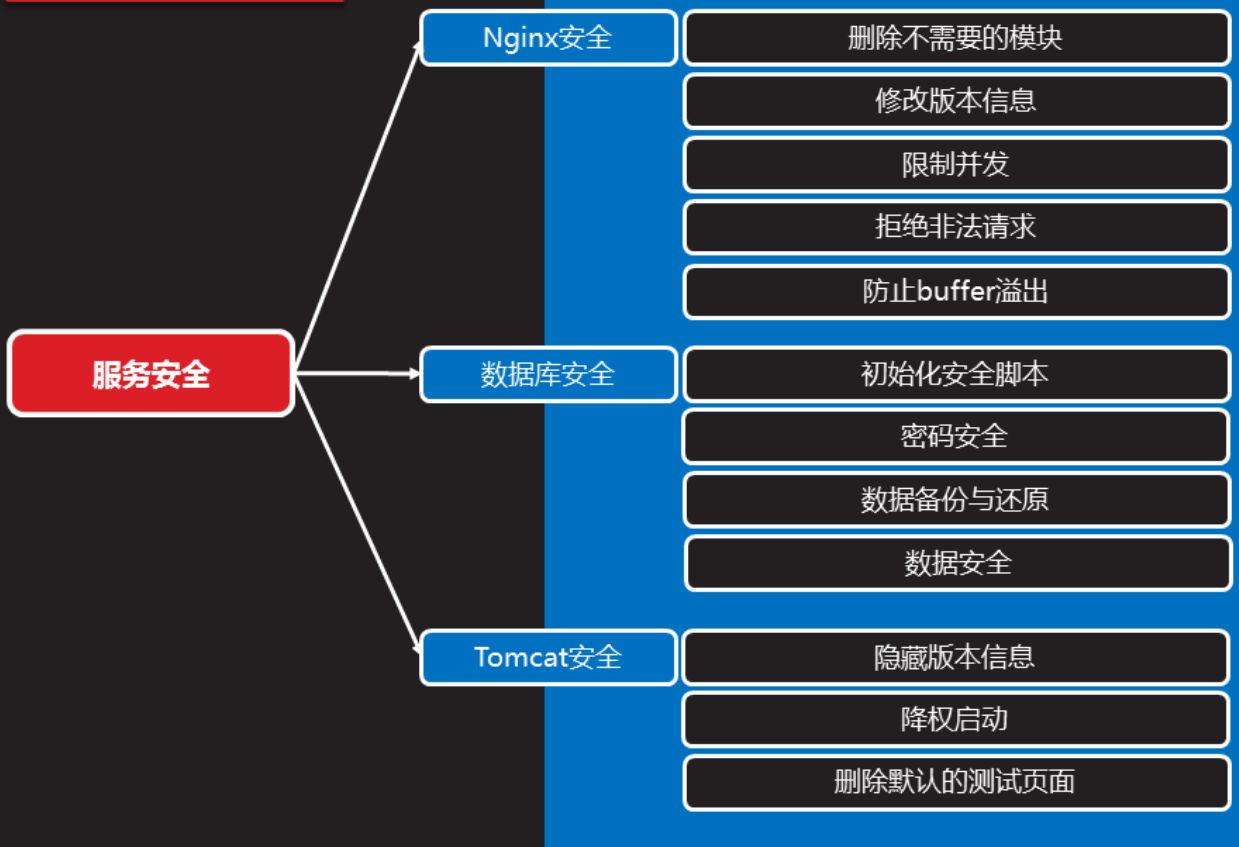
```
[roo@svr7 ~]# ausearch -k sshd_config -i
//根据key搜索日志，-i为交互式操作
```

案例1：部署audit监控文件

课堂练习

1. 使用audit监控/etc/ssh/sshd_config
2. 当该文件发生任何变化即记录日志
3. 通过手动和ausearch工具查看日志内容

服务安全



Tedu.cn 达内教育

Nginx安全

Tedu.cn 达内教育

删除不需要的模块

知识讲解

- Nginx是模块化设计
  - 需要的模块使用--with加载模块
  - 不需要的模块使用--without禁用模块
- 最小化安装永远的对的！！！

```
[roo@svr7 nginx-1.12]# ./configure \
>--without-http_autoindex_module \
>--without-http_ssi_module
[roo@svr7 nginx-1.12]# make
[roo@svr7 nginx-1.12]# make install
```

修改版本信息

- 如何修改版本信息（修改源码）
- 如何隐藏版本号信息（server_tokens off）

知识讲解

[roo@svr7 nginx-1.12]# vim +48 src/http/ngx_http_header_filter_module.c

修改前

```
static u_char ngx_http_server_string[] = "Server: nginx" CRLF;
static u_char ngx_http_server_full_string[] = "Server: " NGINX_VER CRLF;
static u_char ngx_http_server_build_string[] = "Server: " NGINX_VER_BUILD CRLF;
```

修改后

```
static u_char ngx_http_server_string[] = "Server: Jacob" CRLF;
static u_char ngx_http_server_full_string[] = "Server: Jacob" CRLF;
static u_char ngx_http_server_build_string[] = "Server: Jacob" CRLF;
```

限制并发

- ngx_http_limit_req_module为默认模块
  - 该模块可以降低DDos攻击风险

知识讲解

[roo@svr7 ~]# vim /usr/local/nginx/conf/nginx.conf

```
http{
limit_req_zone $binary_remote_addr zone=one:10m rate=1r/s;
        server {
                listen 80;
                server_name localhost;
                limit_req zone=one burst=5;
        }
}
```

限制并发（续1）

Tedu.cn 达内教育

知识讲解

- 下面配置的功能为：
  - 语法：limit_req_zone key zone=name:size rate=rate;
  - 将客户端IP信息存储名称为one的共享内存，空间为10M
  - 1M可以存储8千个IP的信息，10M存8万个主机状态
  - 每秒中仅接受1个请求，多余的放入漏斗
  - 漏斗超过5个则报错

```
limit_req_zone $binary_remote_addr  zone=one:10m rate=1r/s;
limit_req zone=one burst=5;
```

```
[roo@client ~]# ab -c 100 -n 100  http://192.168.4.5/
```

拒绝非法请求

Tedu.cn 达内教育

知识讲解

- 常见HTTP请求方法
  - HTTP定义了很多方法，实际应用中一般仅需要get和post

请求方法	功能描述
GET	**请求指定的页面信息，并返回实体主体**
HEAD	类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头
POST	**向指定资源提交数据进行处理请求（例如提交表单或者上传文件）**
DELETE	请求服务器删除指定的页面
PUT	向服务器特定位置上传资料
…	其他

拒绝非法请求（续1）

Tedu.cn 达内教育

知识讲解

- 禁用其他方法，仅允许(GET|POST)

```
[roo@svr7 ~]# vim /usr/local/nginx/conf/nginx.conf
http{
        server {
                listen 80;
                if ($request_method !~ ^(GET|POST)$ ) {
                        return 444;
                }
        }
}
[roo@client ~]# curl -i -X GET "http://192.168.4.5"
[roo@client ~]# curl -i -X HEAD "http://192.168.4.5"
```

防止buffer溢出

Tedu.cn 达内教育

知识讲解

- 防止客户端请求数据溢出
- 有效降低机器Dos攻击风险

```
[roo@svr7 ~]# vim /usr/local/nginx/conf/nginx.conf
http{
                client_body_buffer_size  1K;
                client_header_buffer_size 1k;
                client_max_body_size 16k;
                large_client_header_buffers 4 4k;
 ... ...
}
```

数据库安全

初始化安全脚本

知识讲解

- 输入就密码，配置新root密码
- Remove anonymous users（删除匿名账户）
- Disallow root login remotely?（禁止root远程登录）
- Remove test database（删除测试数据库）
- Reload privilege（刷新权限）

```
[roo@svr7 ~]# systemctl status mariadb          //确保服务已启动
[roo@svr7 ~]# mysql_secure_installation         //执行初始化安全脚本
```

密码安全

知识讲解

- 修改MySQL密码的若干方法

```
[roo@svr7 ~]# mysqladmin -uroot -predhat password 'mysql'
//修改密码，旧密码为redhat，新密码为mysql
[roo@svr7 ~]# mysql -uroot -pmysql
MariaDB [(none)]>set password for root@'localhost'=password('redhat')
//使用账户登录数据库，修改密码
MariaDB [(none)]> select user,host,password from mysql.user;
+--------------+---------------+-------------------------------------------+
| user         | host          | password                                  |
+--------------+---------------+-------------------------------------------+
| root         | localhost     | *84BB5DF4823DA319BBF86C99624479A198E6EEE9 |
| root         | 127.0.0.1     | *84BB5DF4823DA319BBF86C99624479A198E6EEE9 |
| root         | ::1           | *84BB5DF4823DA319BBF86C99624479A198E6EEE9 |
+--------------+---------------+-------------------------------------------+
```

密码安全（续1）

知识讲解

- 问题是历史记录会出卖你!
- binlog日志里有明文密码（5.6版本后修复了）

```
[roo@svr7 ~]# cat .bash_history
mysqladmin -uroot -pxxx password 'redhat'
[roo@svr7 ~]# cat .mysql_history
set password for root@'localhost'=password('redhat');
select user,host,password from mysql.user;
flush privileges;
```

- 解决：
  - 管理好自己的历史，不使用明文登录，选择合适的版本
  - 日志，行为审计
  - 防火墙从TCP层设置ACL（禁止外网接触数据库）

数据备份与还原

Tedu.cn 达内教育

知识讲解

- 备份

```
[roo@svr7 ~]# mysqldump -uroot -predhat mydb table > table.sql
[roo@svr7 ~]# mysqldump -uroot -predhat mydb > mydb.sql
[roo@svr7 ~]# mysqldump -uroot -predhat --all-databases > all.sql
```

- 还原

```
[roo@svr7 ~]# mysql -uroot -predhat mydb  < table.sql     //还原表
[roo@svr7 ~]# mysql -uroot -predhat mydb  < mydb.sql    //还原数据库
[roo@svr7 ~]# mysql -uroot -predhat < all.sql                 //还原所有
```

数据安全

Tedu.cn 达内教育

知识讲解

- 创建可以远程登录的账户

```
[roo@svr7 ~]# mysql -uroot -predhat
MariaDB [(none)]> grant all on *.* to tom@'%' identified by '123';
```

- 使用tcpdump抓包

```
[roo@svr7 ~]# tcpdump -w log -i eth0 src or dst port 3306
```

- 客户端远程登录数据库，查看抓包数据

```
[roo@client ~]# mysql -utom -p123 -h 192.168.4.5
MariaDB [(none)]> select * from mysql.user;
[roo@svr7 ~]# tcpdump -A -r log
```

- 解决：使用SSL或SSH加密数据传输

Tomcat安全

隐藏版本信息

知识讲解

- 修改tomcat主配置文件，隐藏版本信息

```
[roo@svr7 tomcat]# yum -y install java-1.8.0-openjdk-devel
[roo@svr7 tomcat]# cd lib/ ;  jar -xf catalina.jar
[roo@svr7 tomcat]# vim org/apache/catalina/util/ServerInfo.properties //修改内容
[roo@svr7 tomcat]# vim /usr/local/tomcat/conf/server.xml
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"  redirectPort="8443" server="jacob" />
```

- 测试

```
[roo@svr7 ~]# curl -I http://192.168.2.100:8080/xx      //头部信息
[roo@svr7 ~]# curl -I http://192.168.2.100:8080         //头部信息
[roo@svr7 ~]# curl http://192.168.2.100:8080/xx         //报错页面
```

降权启动

Tedu.cn 达内教育

知识讲解

- 使用非root启动tomcat服务

```
[roo@svr7 ~]# useradd tomcat
[roo@svr7 ~]# chown -R tomcat:tomcat /usr/local/tomcat/
[roo@svr7 ~]# su -c /usr/local/tomcat/bin/startup.sh tomcat
```

- 开机启动

```
[roo@svr7 ~]# chmod +x /etc/rc.local
[roo@svr7 ~]# vim /etc/rc.local                    //添加如下内容
 su -c /usr/local/tomcat/bin/startup.sh tomcat
```

删除默认的测试页面

Tedu.cn 达内教育

知识讲解

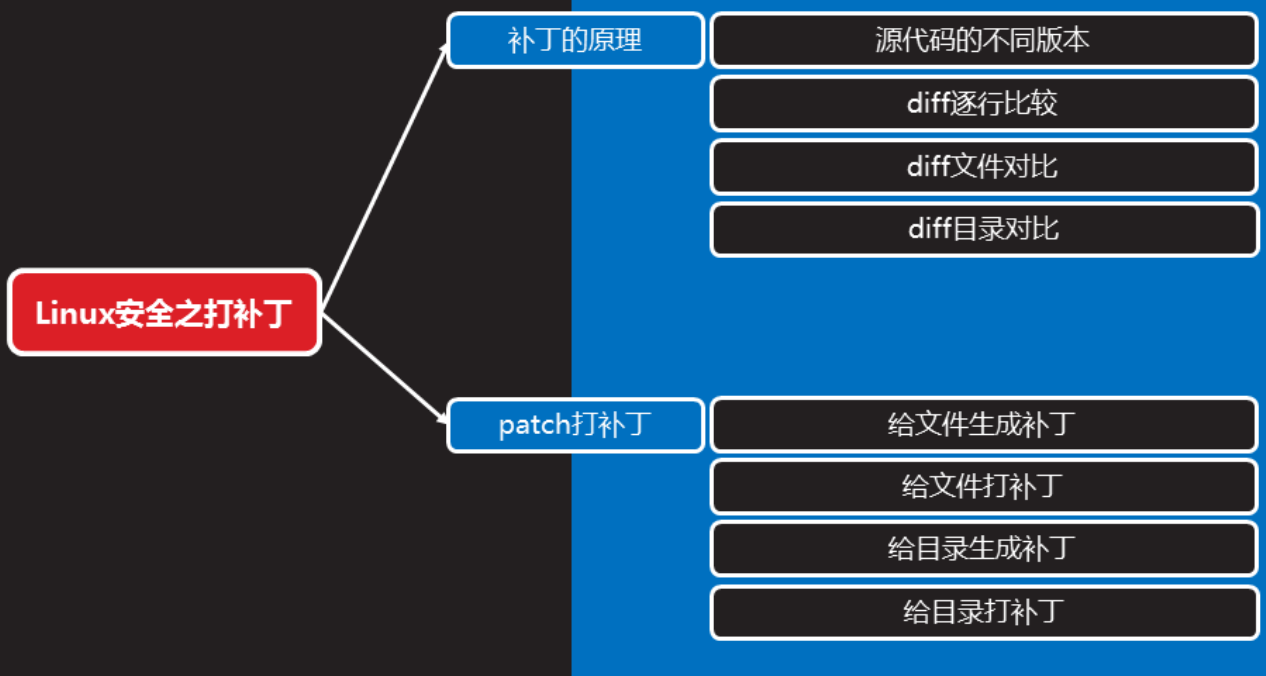
```
[roo@svr7 ~]# rm -rf  /usr/local/tomcat/webapps/*
```

案例2：加固常见服务的安全

课堂练习

1. 优化Nginx服务的安全配置
2. 优化MySQL数据库的安全配置
3. 优化Tomcat的安全配置

Linux安全之打补丁



Tedu.cn 达内教育

补丁的原理

Tedu.cn 达内教育

源代码的不同版本

知识讲解

- V1版本

```
[roo@svr7 ~]# cat test1.sh
#!/bin/bash
echo "hello wrld"
```

- V2版本（修复错误、添加功能）

```
[roo@svr7 ~]# cat test2.sh
#!/bin/bash
echo "hello world"
echo "test file"
```

diff逐行比较

Tedu.cn 达内教育

知识讲解

- diff的原则是：
  - 告诉我们怎么修改第一个文件后能得到第二个文件
- 选项
  - -u　　输出统一内容的头部信息（打补丁使用）
  - -r　　递归对比目录中的所有资源（可以对比目录）
  - -a　　所有文件视为文本（包括二进制程序）
  - -N　　无文件视为空文件（空文件怎么变成第二个文件）

//A目录下没有txt文件，B目录下有txt文件
//diff比较两个目录时，默认会提示txt仅在B目录有（无法根据补丁修复A缺失的文件）
//diff比较时使用N选项，则diff会拿B下的txt与A下的空文件对比
//补丁信息会明确说明如何从空文件修改后变成txt文件，打补丁即可成功！

diff文件对比

Tedu.cn 达内教育

知识讲解

- 仅对文件比较

```
[roo@svr7 demo]# cat test1.sh
#!/bin/bash
echo "hello world"
echo "test"
[roo@svr7 demo]# cat test2.sh
#!/bin/bash
echo "hello the world"
echo "test file"
```

```
[roo@svr7 demo]# diff -u test1.sh test2.sh
--- test1.sh    2018-02-07 22:20:02.723971251 +0800
+++ test2.sh 2018-02-07 22:20:13.358760687 +0800
@@ -1,3 +1,3 @@
 #!/bin/bash
-echo "hello world"
-echo "test"
+echo "hello the world"
+echo "test file"
```

//备注：第一个文件删除前两行后，再把+后面的内容添加上，即可变成第二个文件！

diff目录对比

Tedu.cn 达内教育

知识讲解

- 准备环境

```
[roo@svr7 demo]# mkdir {source1,source2}
[roo@svr7 demo]# echo "hello world"       > source1/test.sh
[roo@svr7 demo]# echo "hello the world" > source2/test.sh
[roo@svr7 demo]# echo "test" > source2/tmp.txt
[roo@svr7 demo]# cp /bin/find source1/
[roo@svr7 demo]# cp /bin/find source2/
[roo@svr7 demo]# echo "1" >> source2/find
```

```
[roo@svr7 demo]#  tree source1/
|-- find
`-- test.sh
```

```
[roo@svr7 demo]#  tree source2/
|-- find
|-- test.sh
`-- tmp.txt
```

diff目录对比（续1）

Tedu.cn 达内教育

知识讲解

- 对比差异

```
[roo@svr7 demo]# diff -u source1/ source2/
//仅对比了文本文件test.sh；二进制文件、tmp都没有对比差异，仅提示。
[roo@svr7 demo]# diff -Nu source1/ source2/
//对比了test.sh，并且使用source2目录的tmp.txt与source1的空文件对比差异。
[roo@svr7 demo]# diff -Nua source1/ source2/
//对比了test.sh、tmp.txt、find(程序)。
```

patch打补丁

给文件生成补丁

知识讲解

- 准备环境，生成补丁文件

```
[roo@svr7 ~]# mkdir demo; cd demo
[roo@svr7 demo]# vim test1.sh
#!/bin/bash
echo "hello world"
echo "test"
[roo@svr7 demo]# vim test2.sh
#!/bin/bash
echo "hello the world"
echo "test file"
```

```
[roo@svr7 demo]# diff -Nua test1.sh test2.sh > test.patch
```

给文件打补丁（续1）

- 对旧版本的代码，使用补丁即可更新，而不需要下载完成的新代码（往往完整的程序很大）

知识讲解

```
[roo@svr7 demo]# yum -y install patch
[roo@svr7 demo]# patch -p0 < test.patch
patching file test1.sh
[roo@svr7 demo]# patch -RE < test.patch        //还原旧版本，反向修复
```

```
//patch -pnum指定删除补丁文件中多少层路径前缀
//如原始路径为/u/howard/src/blurfl/blurfl.c
//-p0则整个路径不变
//-p1则修改路径为u/howard/src/blurfl/blurfl.c
//-p4则修改路径为blurfl/blurfl.c
//-R(reverse)反向修复，-E修复后如果文件为空，则删除该文件
```

给文件打补丁（续2）

- 在不同目录生成补丁文件

知识讲解

```
[roo@svr7 demo]# cd ..                                  //返回上层目录
[roo@svr7 ~]# diff -Nura demo/test1.sh demo/test2.sh > test.patch
[roo@svr7 ~]# cat test.patch                            //补丁文件带路径信息
---   demo/test1.sh          2018-02-08 22:03:14.284229418 +0800
+++ demo/test2.sh          2018-02-07 22:20:13.358760687 +0800
[roo@svr7 ~]# patch -p0 < test.patch                    //在demo外打补丁
[roo@svr7 ~]# cd demo
[roo@svr7 ~]# patch -p1 < ../test.patch                 //在demo内打补丁
```

给目录生成补丁

Tedu.cn 达内教育

- 使用前面创建的目录环境

知识讲解

```
[roo@svr7 demo]#  tree source1/
|-- find
`-- test.sh
```

```
[roo@svr7 demo]#  tree source2/
|-- find
|-- test.sh
`-- tmp.txt
```

```
[roo@svr7 demo]# diff -Nuar source1/ source2/ > source.patch
```

给目录打补丁

Tedu.cn 达内教育

- 对就版本的代码，使用补丁即可更新，而不需要下载完成的新代码（往往完整的程序很大）

知识讲解

```
[roo@svr7 demo]# ls
source1  source2  source.patch
[roo@svr7 demo]# cat source.patch              //对比的文件有路径信息
---  source1/test.sh 2018-02-07 22:51:33.034879417 +0800
+++ source2/test.sh 2018-02-07 22:47:32.531754268 +0800
@@ -1 +1 @@
-hello world
+hello the world
[roo@svr7 demo]# cd source1
[roo@svr7 source1]# patch  -p1 < ../source.patch
```

案例3：使用diff和patch工具打补丁

Tedu.cn 达内教育

- 使用diff对比文件差异
- 使用diff生成补丁文件
- 使用patch命令为旧版本打补丁

课堂练习

总结和答疑

Tedu.cn 达内教育

打补丁错误

Tedu.cn 达内教育

问题现象

- 对目录生成补丁文件后在目录外使用补丁文件打补丁
- 对于新目录下新建的文件，旧目录中没有该文件
- patch无法在旧目录创建文件（修复）

知识讲解

故障分析及排除

Tedu.cn 达内教育

知识讲解

- 原因分析
  - 使用patch命令在目录外打补丁时会提示如下信息：
    The next patch would create the file source2/tmp.txt,
    which already exists! Assume -R? [n]
- 解决办法
  - 进入需要修复、打补丁的目录中，
  - 使用patch -p1 < ../补丁文件，就可以成功修复