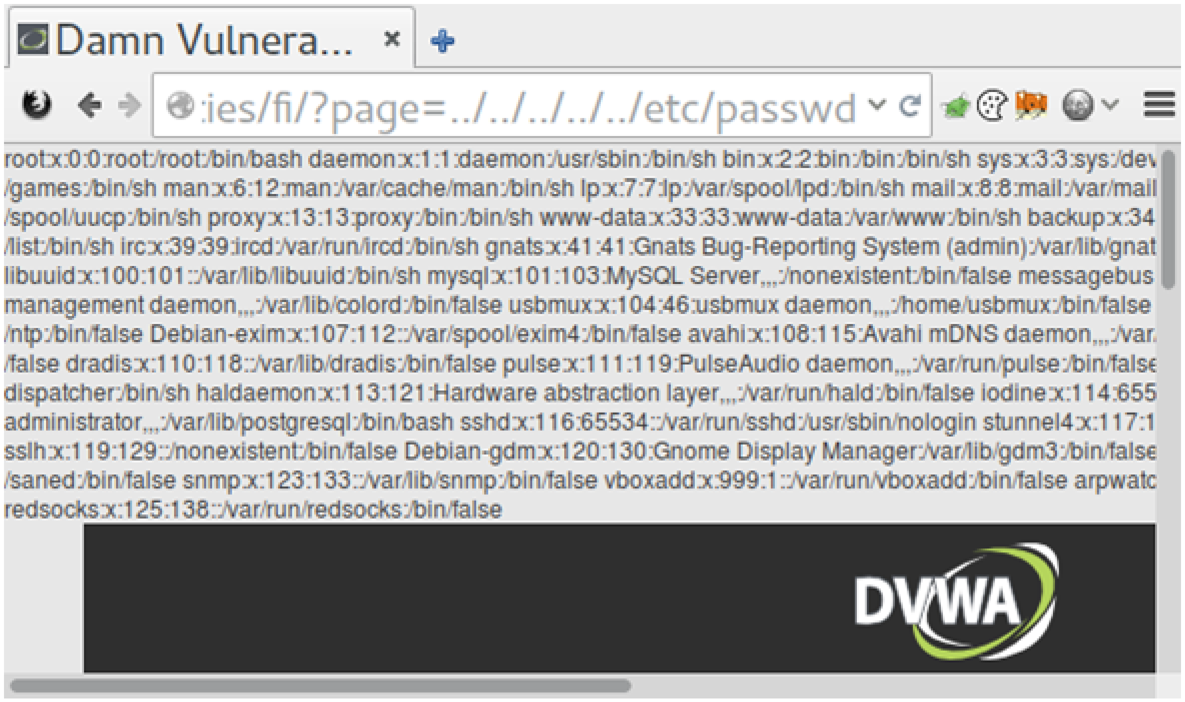
Directory Traversal, File Inclusion, and The Proc File System

Directory traversal and local file inclusion bugs are frequently seen in web applications. Directory traversal is when a server allows an attacker to read a file or directories outside of the normal web server directory. Local file inclusion allows an attacker the ability to include an arbitrary local file (from the web server) in the web server's response. Both of these bugs can be used to read arbitrary files from the server.



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev
/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail
/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34
/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnat
libuuid:x:100:101::/var/lib/libuuid:/bin/sh mysql:x:101:103:MySQL Server,,,:/nonexistent:/bin/false messagebus
management daemon,,,:/var/lib/colord:/bin/false usbmux:x:104:46:usbmux daemon,,,:/home/usbmux:/bin/false
/ntp:/bin/false Debian-exim:x:107:112::/var/spool/exim4:/bin/false avahi:x:108:115:Avahi mDNS daemon,,,:/var.
/false dradis:x:110:118::/var/lib/dradis:/bin/false pulse:x:111:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
dispatcher:/bin/sh haldaemon:x:113:121:Hardware abstraction layer,,,:/var/run/hald:/bin/false iodine:x:114:655
administrator,,,:/var/lib/postgresql:/bin/bash sshd:x:116:65534::/var/run/sshd:/usr/sbin/nologin stunnel4:x:117:1
sslh:x:119:129::/nonexistent:/bin/false Debian-gdm:x:120:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
/saned:/bin/false snmp:x:123:133::/var/lib/snmp:/bin/false vboxadd:x:999:1::/var/run/vboxadd:/bin/false arpwatc
redsocks:x:125:138::/var/run/redsocks:/bin/false
```

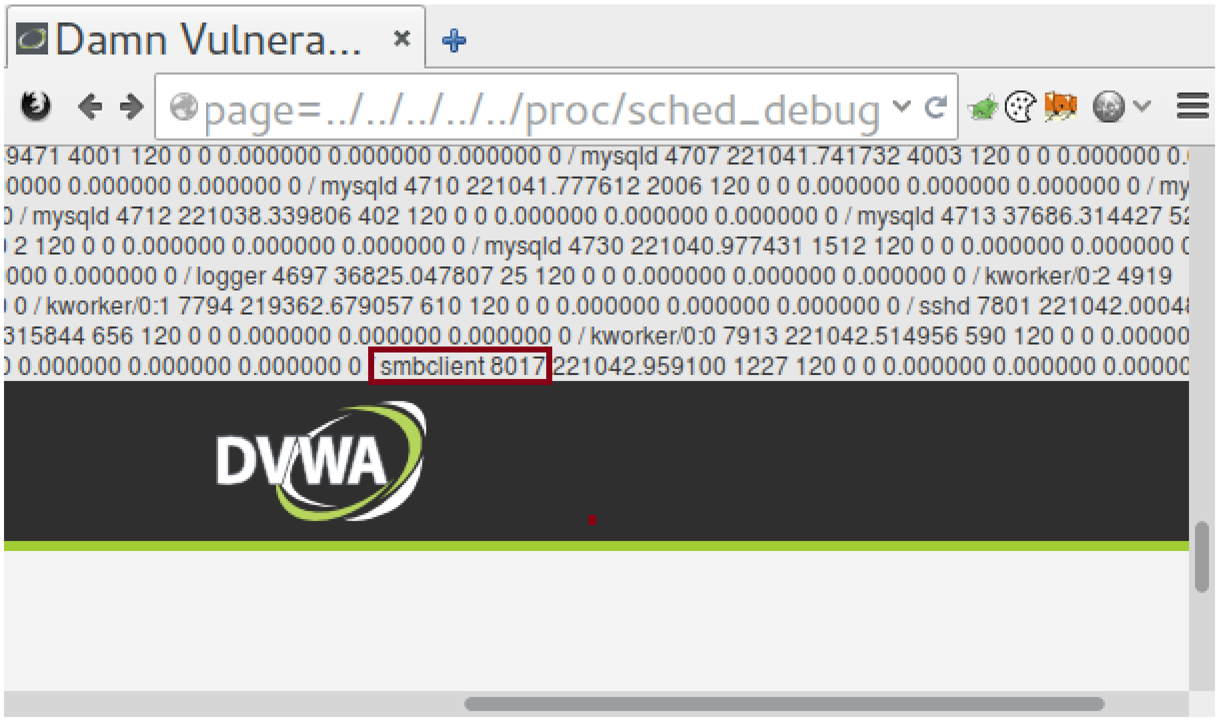
FILE INCLUSION EXAMPLE IN DVWA

In most cases, this means that an attacker can read the /etc/passwd file and the shell history files in order to find information leaks. However, an attacker can also use this to read the proc file system. This can provide some interesting insights into what's running on the server.

A few of the more interesting proc entries include:

Directory	Description
/proc/sched_debug	This is usually enabled on newer systems, such as RHEL 6. It provides information as to what process is running on which cpu. This can be handy to get a list of processes and their PID number.
/proc/mounts	Provides a list of mounted file systems. Can be used to determine where other interesting files might be located
/proc/net/arp	Shows the ARP table. This is one way to find out IP addresses for other internal servers.
/proc/net/route	Shows the routing table information.

/proc/net/tcp and /proc/net/udp	Provides a list of active connections. Can be used to determine what ports are listening on the server
/proc/net/fib_trie	This is used for route caching. This can also be used to determine local IPs, as well as gain a better understanding of the target's networking structure
/proc/version	Shows the kernel version. This can be used to help determine the OS running and the last time it's been fully updated.

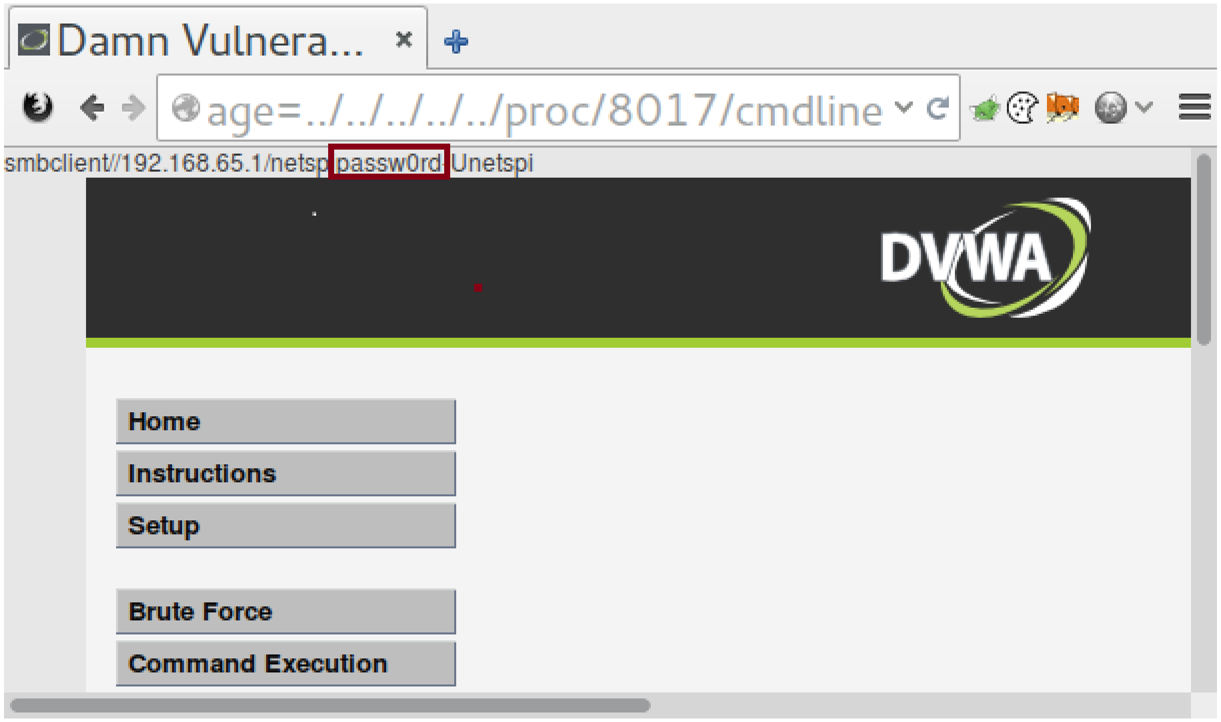


```
9471 4001 120 0 0 0.000000 0.000000 0.000000 0 / mysqld 4707 221041.741732 4003 120 0 0 0.000000 0.
0000 0.000000 0.000000 0 / mysqld 4710 221041.777612 2006 120 0 0 0.000000 0.000000 0.000000 0 / my
0 / mysqld 4712 221038.339806 402 120 0 0 0.000000 0.000000 0.000000 0 / mysqld 4713 37686.314427 52
2 120 0 0 0.000000 0.000000 0.000000 0 / mysqld 4730 221040.977431 1512 120 0 0 0.000000 0.000000 0
000 0.000000 0 / logger 4697 36825.047807 25 120 0 0 0.000000 0.000000 0.000000 0 / kworker/0:2 4919
0 / kworker/0:1 7794 219362.679057 610 120 0 0 0.000000 0.000000 0.000000 0 / sshd 7801 221042.0004
315844 656 120 0 0 0.000000 0.000000 0.000000 0 / kworker/0:0 7913 221042.514956 590 120 0 0 0.00000
0 0.000000 0.000000 0.000000 0 / smbclient 8017 221042.959100 1227 120 0 0 0.000000 0.000000 0.00000
```

OUTPUT OF /PROC/SCHED_DEBUG SHOWS SMBCLIENT RUNNING ON PID 8017

Each process also has its own set of attributes. If you have the PID number and access to that process, then you can obtain some useful information about it, such as its environmental variables and any command line options that were run. Sometimes these include passwords. Linux also has a special proc directory called *self* which can be used to query information about the current process without having to know it's PID. In the following examples you can replace [PID] with either self or the PID of the process you wish to examine.

Directory	**Description**
/proc/[PID]/cmdline	Lists everything that was used to invoke the process. This sometimes contains useful paths to configuration files as well as usernames and passwords.
/proc/[PID]/environ	Lists all the environment variables that were set when the process was invoked. This also sometimes contains useful paths to configuration files as well as usernames and passwords.
/proc/[PID]/cwd	Points to the current working directory of the process. This may be useful if you don't know the absolute path to a configuration file.
/proc/[PID]/fd/[#]	Provides access to the file descriptors being used. In some cases this can be used to read files that are opened by a process.



smbclient//192.168.65.1/netsppassw0rdUnetspi

CMDLINE LEAKS A CIFS SHARE PASSWORD VIA SMBCLIENT

Combining directory traversal and file inclusion vulnerabilities with the proc file system ends up being a great way to gain access to information relating to the running processes on a Linux system. For instance, it's possible to locate database servers by looking at the current connections the machine has. It also makes it easier to fingerprint what software is running on the machines in order to help determine if they are vulnerable to version-specific issues. It's important not to overlook the proc file system when one of these vulnerabilities are found during a penetration test.