Operational Awareness: Supplement

# Notes and References

Because data for the commands who or w come from the file system, you can write your own code to directly query the data. The man page for utmp provides information on how to access the data it provides in C. Here is a sample C program that reads the data from /var/run/utmp and prints it to the screen.

Listing 3-1. C code to retrieve user login data from /var/run/utmp

/\* userlist.c

\*

\* Sample program to query data from /var/run/utmp

\* Compile: gcc userlist.c -o userlist

\* Run: ./userlist

\*/

#include<fcntl.h>

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<utmp.h>

void print\_record\_type(short type){

if(type == EMPTY) printf(" Invalid Record\n");

if(type == RUN\_LVL) printf(" Change in run level\n");

if(type == BOOT\_TIME) printf(" System boot time\n");

if(type == NEW\_TIME) printf(" Time after system clock change\n");

if(type == OLD\_TIME) printf(" Time before system clock change\n");

if(type == INIT\_PROCESS) printf(" Process spawned by init\n");

if(type == LOGIN\_PROCESS) printf(" Session for user login\n");

if(type == USER\_PROCESS) printf(" Normal process\n");

if(type == DEAD\_PROCESS) printf(" Terminated process\n");

}

int main(int agrc, char\* argv[]) {

struct utmp utmp\_entry;

int utmp\_fd;

utmp\_fd = open(UTMP\_FILE, O\_RDONLY);

if(utmp\_fd < 0) {

perror("Error opening utmp file");

exit(1);

}

while( read(utmp\_fd, &utmp\_entry, sizeof(utmp\_entry))){

printf("Log name: %s\n", utmp\_entry.ut\_name);

print\_record\_type(utmp\_entry.ut\_type);

printf(" PID: %i\n", utmp\_entry.ut\_pid);

printf(" TTY: %s\n", utmp\_entry.ut\_line);

printf(" User: %s\n", utmp\_entry.ut\_user);

printf(" Host: %s\n", utmp\_entry.ut\_host);

}

exit(0);

}

Exercises

1. The tool ss is a Linux tool comparable to netstat. Test the tool and the effect of the options -l (listening ports) -a (all ports) -p (process listing) -e (extended information) -i (internal information) -t (TCP) and -u (UDP).
2. Use the Sysinternals tool pslist from the command line to list the running processes and use pskill to kill a process.
3. Compare and contrast TCPLogView <http://www.nirsoft.net/utils/tcp_log_view.html> with Sysinternals TCPView.
4. Wireshark is vulnerable to direct attack. Install Wireshark 1.4.4 on a Windows system, and use the Metasploit module exploit/windows/misc/wireshark\_packet\_dect to gain a shell on the target.
5. Install the Microsoft Message Analyzer[[1]](#footnote-1), available from https://www.microsoft.com/en-us/download/details.aspx?id=44226. Use it to capture packets during a Metasploit attack against a browser using the reverse HTTPS Meterpreter payload. Can you identify the Meterpreter traffic in the packet capture?

1. The default system requirements for Microsoft Message Analyzer includes at least 2 GB of memory. [↑](#footnote-ref-1)