
Loadable Kernel Modules [LKM] Lecture

By [Unknown](#)

HTML by [Mikkkeee](#)

For [Blacksun Research Facility \[BSRF\]](#)

<http://blacksun.box.sk>

LKM file can be downloaded from <http://blacksun.box.sk/mirror/LKM.zip>

```
<sts|> a zip ???
<Mikkkeee> guys the lecture started?
<Hijack> 5k
<mezzano> it's mikkkeee sup man told you I wouldn't miss this one ;)
<SpiderMan> around 5k
<SpiderMan> not large at all
<Syrup> anyone know how juno password algorithm works?
<Phr3k> small
<Mikkkeee> guys the lecture started?
<SpiderMan> not yet Mikkkeee
<SpiderMan> just told everyone to download the file
*** Mikkkeee sets mode: +v unknown
<unknown> No it starts now
<Revelant-Angel> i can read anything all messages are flying passed me
*** Retrieving #bsrf info...
*** Revelant-Angel has quit IRC (Quit:)
<feds> what is it???
<Syrup> yay lecture
*** DarkneSs has joined #bsrf
<Mikkkeee> its a zip unknown set up
*** bulgarinche has joined #bsrf
<Mikkkeee> http://blacksun.box.sk/mirror/LKM.zip
<Phr3k> i dunno what LKM even is
<Mikkkeee> loadable kernel modules
<unknown> I will explain every thing
<freakOVER> don't get excited kids!
<freakOVER> :)
<feds> hey
<feds> me 2!
<Mikkkeee> well someone log this too
<}{exadecimal> LKM = Loadable Kernel Module
*** optimum has quit IRC (Quit: )
<Phr3k> is this lecture aplicable to even beginners
* freakOVER is loggin
<freakOVER> ;)
<}{exadecimal> i'm logging Mikkkeee
<Phr3k> i'm a beginner beginner
<Dustin> obviously
*** wascy has joined #bsrf
<Mikkkeee> -----beginning of lecture-----
<feds> any other celeb like u
*** Mikkkeee sets mode: +m
<Mikkkeee> if anyone wants voice msg an op
*** Mikkkeee sets mode: -m
<unknown> OK lets start
<ControlC> SWEET
*** _jacs- is now known as jacs
<D|GiTaLM0nKe3> k
<wascy> me, pls.
<Mikkkeee> wait who wants voice
<}{exadecimal> do we need a c compiler? theres a c file in that zip
```

```

<sts|> wich kernelversion?
<sts|> 2.2 2.4?
<Dustin> i owuld like voice
<D|GiTaLM0nKe3> I do
<twix> me
<Phr3k> i want voice
<|StYxX|> no one talk for a sec
<wascy> i'd like. thx.
<freakOVER> ouch
<snider> }{exadecimal: If you use linux you have gcc, if you don't use linux the LKM won't compile anyways
*** Mikkkeee sets mode: +v Phr3k
*** Mikkkeee sets mode: +m
<unknown> you need GCC
<Mikkkeee> unknown you can begin
<unknown> LKM(loadable kernel modules) are used to increase
<unknown> the kernel's functionality on run time.
*** Mikkkeee sets mode: +v mezzano
<unknown> which means you dont have to recompile the entire kernel to
<unknown> use load it.
<unknown> This is why it is used for many device drivers.
<unknown> Because LKM are used as part of the kernel it
<unknown> give us endless ways to use it for our benifits :)
*** SpiderMan has quit IRC (Ping timeout: 180 seconds)
<freakOVER> are LKMs slower to load?
*** SpiderMan has joined #bsrf
*** ChanServ sets mode: +o SpiderMan
<unknown> Every LKM contains 2 main functions
*** h4x0r3d has joined #bsrf
<unknown> int init_module(void) and cleanup_module
*** jacs has joined #bsrf
*** _sokrates- has quit IRC (Quit: Leaving)
<unknown> init_module is called when the module is loaded and cleanup_module
<unknown> is called when you remove it. init_module should return 0
<unknown> to signal that everything is OK.
<unknown> e also must define MODULE and include linux/module.h for
<unknown> module configurations,definitions,macros etc.
*** Phish has joined #bsrf
<unknown> o now we can code our first helloworld LKM
<unknown> look at helloworld.c see of you understand everything.
<unknown> (it should be easy).
<unknown> If u wonder why i used printk and not printf its
<unknown> because we are running in kernel mode which
<unknown> means we can use restricted set of functions
<unknown> A list of useful funcs is in func_list.txt.
*** D|GiTaLM0nKe3 has left #bsrf
<unknown> questions ?
*** snider sets mode: -m
<mezzano> that was deep ;)
<freakOVER> ya
<freakOVER> are LKMs slower to load up than if u just compiled the kernel with them?
<freakOVER> hey it's just a question :P
<unknown> You load it into lernel space in run-time
<snider> feds: i'll send the files
<sts|> you load them.. then they run like they were compiled into the kernel
<freakOVER> ah ok
<feds> i have them
<Hijack> passwords can be captured using the LKM
<unknown> Yes
<Hijack> have them sent through e-mail
<unknown> you can do almost everything using LKM
<}{exadecimal> so how do we use the lkm?
<unknown> OK
<feds> it says it can't find DEVCPPE.exe
<unknown> So let continue
<suspect> lol
<Mikkkeee> unknown would these files run as vxd files?
<Dustin> unknown: but they would need to RUN your lkm first to steal passes?

```

<SpiderMan> feds: just open it up in notepad for now
<unknown> So lets continue
*** snider sets mode: +m
<freakOVER> feds: open it using notepad
<unknown> You can also hack the sever and load it yourself
<unknown> It almost like VxD but in linux
<Mikkkeee> are you going to give us an example in real time?
<unknown> Yes
<Mikkkeee> man vxd hacking is hot
*** Dustin has left #bsrf
<unknown> So compile helloworld.c with gcc -O3 -c helloworld.c
*** pitpat has joined #bsrf
<unknown> This will create helloworld.o
*** Syrup has quit IRC (Quit:)
<unknown> to insert the LKM do "insmod helloworld.o"
*** dg-2 has quit IRC (Quit:)
<unknown> Now do lsmod
<unknown> this will list the modules loaded into the kernel
<unknown> Later we will see how to make our LKM
<unknown> Unremovabe & invisible
<unknown> to remove it do "rmmod helloworld"
<unknown> questions?
*** DG-2 has joined #bsrf
<unknown> questions?
<mezzano> nope seems pretty intuitive ;)
<unknown> KO
<unknown> OK
*** SpiderMan sets mode: -m
<Hijack> so is there any solutions for removing it if i were a system admin
<}{exadecimal> no voice
<Hijack> ?
<unknown> OK now because our LKM is running in kernel space
<unknown> we cant access user space memory.
<unknown> So lets say one of our functions gets as
<unknown> an argument a pointer to user space
<unknown> we must move it to kernel space before we try
<unknown> to use data it points to.
<_miketr0nix-> hello
*** SpiderMan sets mode: +m
<unknown> to use data it points to.
<unknown> He can remove it on run-time he can restart the box
<unknown> so the module will not be loaded
*** Mikkkeee sets mode: +o unknown
*** ChanServ sets mode: -o unknown
<Mikkkeee> shit
<unknown> but first he has to find it
* mezzano thinks mikkkeee needs to pimp-slap chanserv ;)
<unknown> If you what you can put some script to insmod it when the box starts
<unknown> In order to move user data to kernel data we have several functions
*** ControlC has quit IRC (Quit:)
<unknown> get_user(kernel_dest,user_pointer) - gets the value user_pointer points to
<unknown> and put it in kernel_dest(old kernels use dest=get_user(src)).
<unknown> put_user(kernel_src,user_pointer) - note that kernel_src
<unknown> is not a pointer)
<unknown> copy_from_user(kernel_dest_ptr,src_ptr,size)
<unknown> copy_to_user(user_dest_ptr,src_ptr,size)
*** DG-2 has quit IRC (Quit: Leaving)
*** pitpat has quit IRC (Quit:)
*** Mikkkeee sets mode: +o unknown
*** optimum has joined #bsrf
<unknown> For example look at _h_lkm.c at line 78
*** suspect has quit IRC (Ping timeout: 180 seconds)
<unknown> this function copies string from user
*** Dustin has joined #bsrf
<unknown> (GET_U is a macro for get_user())
<unknown> questions ?
*** [phRoZen] has joined #BSRF

```
*** SpiderMan sets mode: -m
<Hijack> none at the moment
*** optimum is now known as optimum[d
<Neophyte> im fine
<}{exadecimal> yeah
*** zaxil has joined #bsrf
<zaxil> hey everyone
<}{exadecimal> why do we keep losing our voices?
*** bulgarinche has quit IRC (Ping timeout: 180 seconds )
<unknown> OK lets continue
<Neophyte> he sets moderation
<Mikkkeee> cause you interrupt
*** optimum[d is now known as optimum[downloading]
<i-o> the printk, where can i see it??
<}{exadecimal> i know what he does but why
*** Mikkkeee sets mode: +m
<unknown> Moving on to symbol table.
<unknown> The symbol table is a file found in /proc/ksyms which contains exported
<unknown> kernel symbols(functions for example) that can be used
<unknown> by our LKM and kernel.
<unknown> The reason we should care about this file is that it will also contain
<unknown> our LKM functions so a good admin will be able to spot
<unknown> our LKM. It can also be used by us to spot and admin LKM
*** Drager has quit IRC (Ping timeout: 180 seconds )
<unknown> If he monitors the entire system with some super LKM
<unknown> we should reconsider our actions).
<unknown> BTW If you encounter and detect this kind of LKM try
<unknown> to DoS the machine and make it restart many admins
<unknown> dont activate the LKM's on init.
*** count_chocula has joined #bsrf
<unknown> Anyway to control the exported symbols all have to do is to
<unknown> is to use macros defined in module.h (in old kernel
<unknown> we need to include /linux/symtab_begin.h and
<unknown> use register_symtab(NULL) if we dont want to
<unknown> export symbols).
*** _Altus- has joined #bsrf
<unknown> The macros are 'EXPORT_NO_SYMBOLS' if dont want to export
<unknown> symbols.
<unknown> If we want to export something define EXPORT_SYMTAB
<unknown> and use 'EXPORT_SYMBOL(name)' macro.
<unknown> questions?
*** Mikkkeee sets mode: -m
<feds> is it works on win kernel?
<unknown> No
<Mikkkeee> no
<SpiderMan> no, for windows use VxDs
<feds> dam
<Mikkkeee> vxds can be discovered
<feds> where do i get it?
<mezzano> strictly *nix baby ;)
<Hijack> the same manner can be applied in window s vxds
<Dustin> vxd == lkm for win?
<unknown> OK lets continue
<Mikkkeee> yah
<Hijack> yes , very true
<feds> where can i get it?
*** Mikkkeee sets mode: +m
<unknown> Now the interesting part - system calls
<unknown> Every OS has built in functions that are used for every operation
<unknown> on the system in linux those functions are called system calls.
<unknown> So if we control the system calls we can control the entire
<unknown> system.
<unknown> To see a list of all the system calls look in
<unknown> look in /usr/include/bits/syscall.h(or run a search for
<unknown> syscall.h).
<unknown> so lets say we want to find which system call
<unknown> is used to list files(which system call is used by ls).
```

<unknown> Do 'strace ls'.
<unknown> strace will give you all the system calls used by ls.
<unknown> Most of the functions are obvious and many repeat many times
<unknown> but look at getdents it shows only twice and the second
<unknown> argument stats "*" entries" if you check this is
<unknown> the number of files + folders + hidden files
<unknown> in the folder.
<unknown> * is a number
<unknown> So we can assume that this is the system call.
<unknown> Now the next step is to try to 'man getdents'.
<unknown> Getdents has manual entry so it explains what it
<unknown> does and we were right :)
*** Norton has joined #bsrf
<unknown> If there isnt man entry you can search the kernel sources
<unknown> (you should always search the sources to see how something works
<unknown> before replacing it).
<unknown> Next we will replace it and modify it to hide every file or folder
<unknown> that starts with _h_.
<unknown> questions?
*** SpiderMan sets mode: -m
<_miketr0nix-> hi
<Dustin> yes
<_miketr0nix-> i have a question
<Dustin> getdents is a program like strace?
*** XarZ has quit IRC (Ping timeout: 180 seconds)
<_miketr0nix-> i want to know things about how to stop my ping
<unknown> getdents is a system call
<_miketr0nix-> but i am in windows
<Mikkkeee> stop your ping?
<_miketr0nix-> yes
<zaxil> get a firewall to blok them
<_miketr0nix-> like someone is trying to ping me
<Dustin> lets save this for another time, shall we?
<unknown> strace can be used to out which functions are used
*** _miketr0nix- has quit IRC (Quit:)
<Norton> heh
<unknown> strace can be used to fing out which functions are used
<Hijack> unknown
*** fatboyjoe has joined #bsrf
<snider> i have a q..
<unknown> by a program
<Dustin> then what is getdents for
<Hijack> one question before i leave , which is not related to LKM
<unknown> getdents is used to get a list of files and folder from
<unknown> a directory
<unknown> ls uses it to get its data
<snider> unknown: i use kernel 2.4.1 which is too old to load that helloworld module.. how come? how can the system see that
its been written for a newer kernel version?
<Dustin> oh ls uses a syscall called 'getdents'?
<Hijack> will the topic ACK Tunnelling be brought up in future lectures?
<Mikkkeee> yah
*** zaxil has quit IRC (Quit:)
<unknown> Dustin: yes
<Dustin> ok, it becomes clear
<Mikkkeee> hijack maybe in the later weeks
<unknown> So if we will replace getdents system call we can hide files
<Hijack> thanks mikkkeee , it s the second attack which i fear most behind LKM / VxD hacking
<Mikkkeee> hehe ack tunneling
*** _fatboyjoe- has quit IRC (Ping timeout: 180 seconds)
*** Hijack has quit IRC (Quit:)
<unknown> and process (since processes are listed in /proc)
<unknown> and processes (since processes are listed in /proc)
<h4x0r3d> heh
<unknown> So lets continue
<Dustin> oh, so getdents() could be programmed to leave out files that begin in !
<unknown> yes
<Dustin> ok, plz continue

```

<unknown> thats our purpose
*** Mikkkeee sets mode: +m
<unknown> To intercept system calls you need to know that the kernel
<unknown> exports a table that lists all the system calls.
<unknown> The var is 'void *sys_call_table[]'.
<unknown> Each entry in this array is a pointer to a system call.
<unknown> You can look in syscall.h to find its number.
<unknown> So if we want to replace getdents system call we can
<unknown> use this code:
<unknown> ...
<unknown> extern void *sys_call_table[];
<unknown> int (*o_getdents)(unsigned int,struct dirent *,unsigned int);
<unknown> int h_getdents("same vars"){
<unknown>     your code
<unknown> }
<unknown> ...
<unknown> int init_module(){
<unknown> ...
<unknown> o_getdents = sys_call_table[SYS_getdents];
<unknown> sys_call_table[SYS_getdents] = h_getdents;
<Phr3k> excellent
<unknown> ...
<unknown> }
<unknown> void cleanup_module(){
<unknown> ...
<unknown> sys_call_table[SYS_getdents] = o_getdents;
<unknown> ...
<unknown> }
<unknown> We need to save the original system call because we might
<unknown> use it and because we need to restore it when
<unknown> we remove the module (unless ou want to fuck up the system
<unknown> Questions?
*** SpiderMan sets mode: -m
<mezzano> so after loading this LKM then all susequent file listing requests would exclude those we don't want them to see
then correct?
<unknown> Yes if code it properly
<unknown> Yes if u code it properly
<Dustin> this is all well and good, but wouldnt you need root access to load a lkm?
<Dustin> and if you had root whqats the point
<snider> what is the purpose of   o_getdents = sys_call_table[SYS_getdents]; ?
<Dustin> of going thru all the trouble
<freakOVER> well if u had gotten into a system
<freakOVER> for like
<freakOVER> a backdoor
<unknown> Dustin: Lets say you want to put a sniffer
<snider> oh.. nevermind me
<unknown> on the system and make it totally invisible
*** JamesBONG has joined #bsrf
<Dustin> ok point taken
<Dustin> oyu cant to that with a program cause a task mgr would list it
<freakOVER> mmhmm
<JamesBONG> does anybody here use macs?
<unknown> what?
<h4x0r3d> lol
<Mikkkeee> hehe no
<freakOVER> $ jobs
<freakOVER> maybe might list it
<Dustin> yeah
*** rs has quit IRC (Ping timeout: 180 seconds )
<freakOVER> or another prog that lists tasks
<freakOVER> LKMs aren't listed as jobs right?
<Dustin> top
<Dustin> ps
<freakOVER> just processes maybe?
*** Norton has quit IRC (Ping timeout: 180 seconds )
<unknown> top and ps wont list it
<Dustin> why not

```

<Mikkkeee> what can list them then
<snider> because its not a userspace program
<Mikkkeee> third party program?
<snider> lsmod lists the loaded kernel modules
<freakOVER> ahh
<Dustin> wait, are we talking about a lkm sniffer or a 'real' sniffer
<unknown> Because we will make the kernel do it
<Dustin> ok
<unknown> i will get to that later
<freakOVER> ok ok i think i'm gettin' this
<unknown> a real sniffer
<Dustin> so to disguise from lsmod you would put the lkm sniffer piggybacked with getdents or something
<unknown> What i will do is to replace getdents and query_module system calls
*** suspect has joined #bsrf
<unknown> so they will remove the what we want from the list
<unknown> OK?
<Dustin> you do need root access first to do all this right?
<Dustin> ok
<snider> yeah, but then after you can hide files from the *right* root of the box
<Mikkkeee> yup
<Dustin> ok
<freakOVER> heehee
<freakOVER> deviousness!
<unknown> You can also spread it as a virus
<unknown> but will get to that
*** snider sets mode: +m
<unknown> SO lets continue
<Mikkkeee> wait can you write a worm to do this for u?
*** Guy_SJS has joined #bsrf
*** ChanServ sets mode: +o Guy_SJS
<unknown> you can intercept DCC or FTP sessions
<unknown> and add code for executables
<unknown> so it will be loaded when the file executed by root
*** SpiderMan has quit IRC (Ping timeout: 180 seconds)
*** DElTa_SquaD has joined #bsrf
*** SpiderMan has joined #bsrf
*** ChanServ sets mode: +o SpiderMan
<unknown> u can c how to hide files on line 172 in _h_lkm.c
*** dataholic has joined #bsrf
*** ChanServ sets mode: +v dataholic
<unknown> Some of the code there is used to hide processes i will explain it later
<unknown> I also replaced chdir system call so in order
<unknown> to execute or read files from hidden folders
<unknown> you must give full path.
<unknown> Try to read it if you got questions ask
<unknown> I commented most of the code so it will be easy to understand
<unknown> questions?
<Phr3k> i gotta go
<mezzano> none yet...the code is pretty easy to read <IMHO>
<Phr3k> someone send me log later
*** Phr3k has quit IRC (Quit:)
<unknown> OK so lets continue
<dataholic> wait wait
* dataholic lights up a cigg
* dataholic puts on a nice mp3
<dataholic> ok go with tha flow!
<dataholic> :)
<unknown> As you know /proc contains a folder named with
<unknown> the process id for each process.
<unknown> To list process with 'ps' for example
<unknown> ps goes through the /proc dir.
<unknown> So If we want to hide a process
<unknown> we check if a process goes through /proc
<unknown> folder so we need to look for the pid of the process we want to hide
<unknown> and remove from the list getdents returns.
<unknown> :)
* Guy_SJS bums a cig off dataholic

<dataholic> lol
<Mikkkeee> heh
<unknown> the /proc folder contains PID's
<unknown> To do it we need to get the inode in which getdents
<unknown> is suppose to scan and if its /proc recover the task name of
<unknown> each pid and if it starts with _h_ remove it from the list.
<unknown> To get the inode we will use the 'current' pointer
<unknown> which points to data of the current process (that issued the call)
<unknown> and its definition is 'struct task_struct *current'(to see
<unknown> struct task_struct look in linux/sched.h).
<unknown> PROC_ROOT_INO is the inode of /proc and its defined in linux/proc_fs.h.
<unknown> The MAJOR and MINOR macros are used to check if it the right device
<unknown> I will explain more about major and minor numbers in
<unknown> the 'coding LKM devices' lecture.
<unknown> Go through the source if you got questions ask.
*** DElTa_SquaD has joined #bsrf
<unknown> Questions?
<Guy_SJS> ok
*** Mikkkeee sets mode: -m
<Guy_SJS> wow
<Guy_SJS> long lecture
<Dustin> can you kill a process by rmdir /proc/911
<h4x0r3d> when will the 'coding LKM devices' be?
<Mikkkeee> next week
<Mikkkeee> maybe
<h4x0r3d> ok, =]
<Dustin> if rmdir is a cmd ;0
<h4x0r3d> heh
*** DElTa_SquaD has left #bsrf
<Dustin> rm -r
<h4x0r3d> well, thanx man, ima go now, peace all!
<dataholic> y'all evil minded ppl :)))
*** h4x0r3d has quit IRC (Quit: i luv box.sk)
<Dustin> suckup
<Mikkkeee> lol
<Guy_SJS> rm -rf /home/dataholc
<freakOVER> he loves the server
<freakOVER> i wanna hit him :(
<Guy_SJS> hope u had a backup.
<dataholic> lol
<dataholic> to dev/null with you!
<unknown> OK lets continue
<Guy_SJS> lol
*** mezzano has quit IRC (Quit:)
*** Mikkkeee sets mode: +m
<unknown> so we have hidden our files,folders and processes
<unknown> but the admin can still see our lkm using lsmod
<unknown> and remove it. There are several methods to
<unknown> hide our modules.
<unknown> One of them (suggested in Phrack 52) is
<unknown> to manipulate struct module and set the size,refs and
<unknown> the name of the module to 0. But when i tried it
<unknown> it didnt work(I think it works only with old kernel,
<unknown> other wise the kernel is loaded but when you lsmod
<unknown> some errors occur).
<unknown> Another way is to intercept sys_query_module system call
<unknown> you can strace lsmod to see it.
<unknown> Its easy to implement(almost like getdents).
<unknown> Look at it in line 321 in _h_lkm.c.
<unknown> To make it unremovable look intercept int delete_module(char *name)
<unknown> and if name is the matches the LKM name return -ENOENT
<unknown> which means the module is not loaded.
<unknown> Questions?
<snider> brilliant
*** snider sets mode: -m
<Dustin> devious little schemer you
<unknown> No questions? so lets continue

<freakOVER> RAWR
<freakOVER> i'm hungry :(
<freakOVER> but linux is sexier :\
<dataholic> lol
*** Mikkkeee sets mode: +m
<unknown> Another cool idea from phrack is redirection of execve
<unknown> for example let say you what that every
<unknown> time login is executed it will execute you login instead
<unknown> so what you is replacing execve and each time login is
<unknown> called you call /hiddenpath/my_login.
<unknown> But ididnt implement it in this module
<unknown> A few notes about execve.
<unknown> To replace execve you must put it in some other
<unknown> place in the sys_call_table(I put it in the end
<unknown> but you can loop and look for the first NULL)
<unknown> and modify the registers before calling it
<unknown> since it expects the data in specific registers.
<unknown> I wrote an asm code to do it in my LKM.
<unknown> If anyone knows a better way to do it please
<unknown> share.
<unknown> I dont recall which line
<unknown> look for it
<snider> 260
<unknown> What it does is wait till sh is executed and if the last
<unknown> argument is ___h___ then it gives you root
<unknown> Someother ideas could be preventing SYS_write from
<unknown> writing certain IPs, make your file
<unknown> totaly unaccessable by replacing open system calls
<unknown> unless you have magic uid, record every thing
<unknown> the admin does and many other things be creative!!!
<unknown> TTY hijacking is a cool thing to do.
<unknown> (I will show how to it in my next lecture)
<unknown> Questions?
*** snider sets mode: -m
<jacs> l
<snider> i guess not
*** stenas has joined #bsrf
<unknown> another important system call is socket(int call,unsigned long *args)).
<unknown> I didnt implemented it due to a lack of time but i will
<unknown> explain how it works.
<unknown> The socket system call handles all the socket operations
<unknown> (recv,send,connect etc.). Its call argument is the type
<unknown> of action.
<unknown> One thing you can do with it is to intercept
<unknown> RECVFROM call and wait for a packet with
<unknown> magic size and magic content and then
<unknown> open a shell on some port to get root.
<unknown> This idea is implemented by plaguez in Phrack 52.
<unknown> But this kind of backdoor will not work
<unknown> if there a firewall on the system.
<Dustin> unless oyu have physical access
*** royanee has quit IRC (Quit:)
<unknown> Sure
<unknown> but you can overcome this
*** stenas is now known as royanee
<unknown> For example you can look for packet that contain
<unknown> some string lets say <123Order321>
<unknown> and remove it from the stream
*** insulted has joined #bsrf
<unknown> This method
<insulted> hi
<unknown> can work without open connection
<Dustin> ./mode #bsrf +m
<unknown> because it will be removed from the stream anyway
*** wascy is now known as wascy[at|work|even|s
<}{exadecimal> gtg, send me a log
<unknown> you can hide the ip of these packets by replacing the write system call

<Mikkkeee> log will be on the bsrf site
<Neophyte> 5-1
*** SpiderMan sets mode: +m
<unknown> OK that it for now if got questions about the source
<unknown> ask
*** SpiderMan sets mode: -m
<snider> i have a kernel too old to load that helloworld module.. how can the system see that its been written for a newer kernel version (even when it's as simple as helloworld.c)?
<snider> thats my only question
<Dustin> it uses a call not implemented in older kernels?
*** CoolMoDee has joined #bsrf
<unknown> What is the messages?
<unknown> What is the message?
<snider> that the module was compiled for.. oh wait.. i replaced my kernel src with the 2.4.7 one a while back
<Mikkkeee> unknown would this work for bsd boxes?
<snider> hehe.. nevermind
*** Saito has joined #bsrf
<unknown> OK so you understood every thing?
<unknown> I didnt try but i shouldnt be hard
*** BootERROR has joined #bsrf
<snider> okay, so the lecture is at end=
<unknown> Well yes
--- End of Lecture---