Win-Trap User Manual

Protection Against Malicious Programs And Code

Without Signature And Complicated Rules

PH Security
www.phsecurity.com
164 Equestrian Drive
Ottawa, Ontario K2M 2B9
Canada
(613) 254-5747

PH Security
164 Equestrian Drive
Ottawa, Ontario K2M 2B9
Canada
http://www.phsecurity.com

Table Of Content

Table Of Figures

Chapter 1 Introduction

First of all, thank you very much for having purchased or evaluating Win-Trap product from PH Security. We hope that you have a wonderful experience with Win-Trap, a last defense line against computer worms such as Code Red, SQL Slammer, and against hackers' code such as WebDAV exploitation, and against email viruses such as W32.Bugbear etc.

1.1 Win-Trap, A Break-Through Research Product

Win-Trap is a product developed by PH Security. It is based on break-through research on the exploitation of buffer overflow vulnerabilities and on email viruses. It provides protection against viruses and worms, without signature and complicated rules, for applications on Windows NT, Windows 2000, Windows XP and Windows Server 2003.

It has two distinct functionalities against malicious code exploiting buffer overflow vulnerabilities and against malicious programs such as email viruses and Trojan programs, as Figure 1-1 shows. The primary functionality of Win-Trap is against malicious code exploiting buffer overflow vulnerabilities.

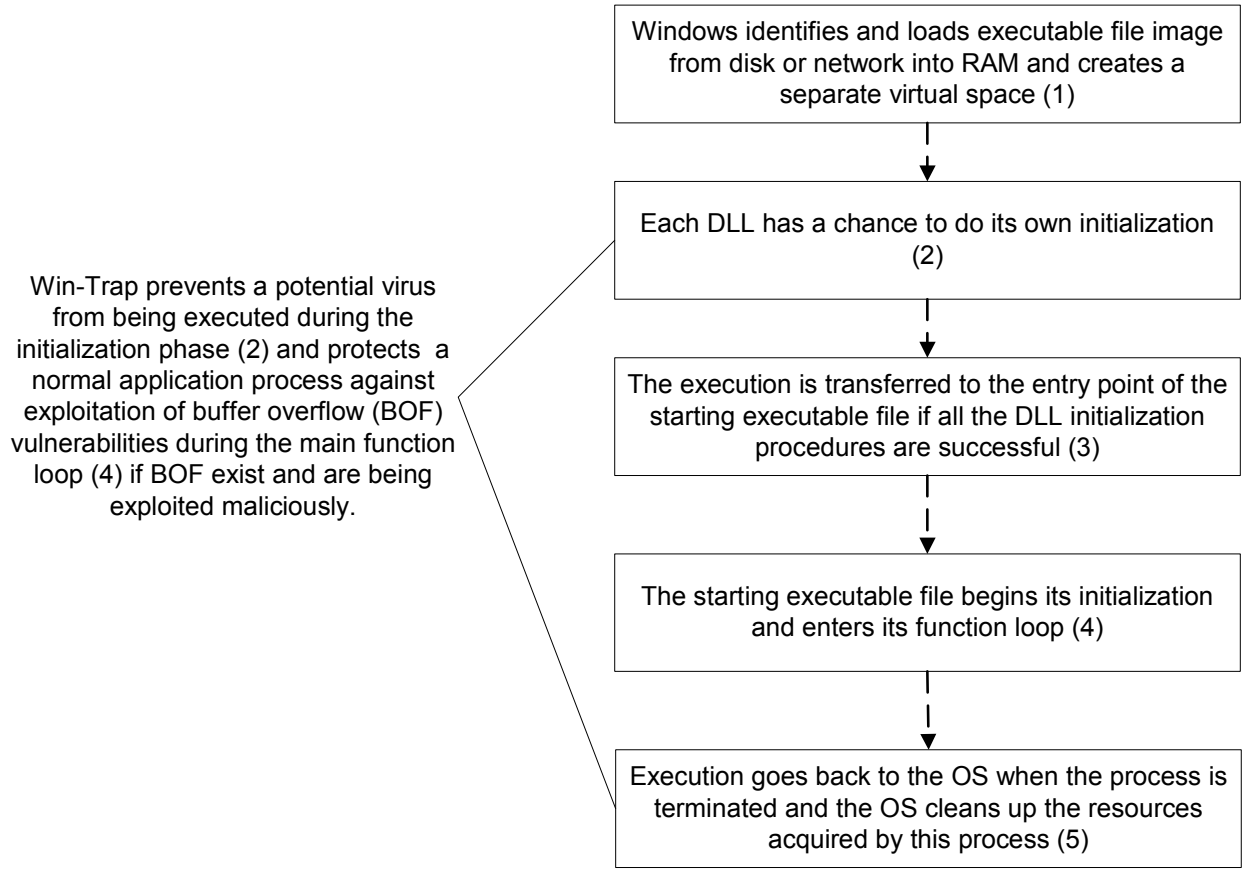


Figure 1-1 Win-Trap provides protection against virus and worm

- When an executable file is being initialized for execution, the Win-Trap protection mechanism verifies whether the executable file is a potential email virus. If it is a probable virus, then the initialization process will be terminated to spare all the damages to be flicked

on by the virus if it is executed. This verification does not require any update of virus signature. Win-Trap does not protect you against script viruses/worms.

- Win-Trap provides run-time protection against exploitation of buffer overflow vulnerabilities for a running application process. Again, this protection mechanism does not require update of worm signature or complicated rules. Win-Trap is the last intrusion detection system (IDS) defense line.

When an abnormal event happens, the Win-Trap will log where and why the event happens so that users can investigate and take corresponding actions with the GUI tools provided in the product.

1.2 Performance Impact Of Win-Trap

- None to unnoticeable. Win-Trap is installed on a Pentium 166 MHz computer with only 80MB RAM, the operation is as normal as it is without the Win-Trap installed.
- It does not affect initialization of normal application programs and does not impact run-time performance of applications.
- Use less than 0.1% of CPU modern CPU resources.

1.3 RAM And Other Requirements

- Less than 256KB RAM for run-time protection
- Less than 5MB of disk installation space, and
- 32-Bit x86 single CPU (Testing on Symmetric Multiple Processor and 64-bit machines in planning)
- Windows NT, 2000, XP and Windows Server 2003

1.4 To Install Win-Trap, Why?

The protection mechanism in Win-Trap is not based on virus and worm signatures as many products on the market are, therefore Win-Trap will likely prevent a brand-new email virus from executing and protect against exploitation of buffer overflow vulnerabilities if they are unable to be patched in a timely fashion. Win-Trap is very effective to halt malicious exploitation of buffer overflow vulnerabilities and malicious programs such as email viruses.

In addition, the protection mechanism in Win-Trap efficiently differentiates between normal code and malicious code, and is not based on setting complicated rules for each program. Therefore, it is very easy to use and analyze an event logging.

Win-Trap complements the current anti-virus methods you deploy and Intrusion Detection System (IDS) you have on your networks and is a very effective and efficient protection mechanism against viruses and worms for Windows for normal users and mission-critical servers alike.

1.5 Lab Testing With Win-Trap Protection

- Latest W32.Bugbear.B, W32.Sobig.A, W32.Sobig.B, W32.Sobig.C, Win32.Sobig.E viruses received through email were double-clicked to execute, they were terminated by Win-Trap in

the initialization process. Actually, some AV software could not even recognize Win32.Sobig.E as a virus yet when the Win-Trap halted its execution!

- Win-Trap halts SQL Slammer-type exploit of SQL Server on Windows 4.0 and Windows 2000;
- Win-Trap halts Code-Red-type exploit of Internet Information Services (IIS) on Windows 2000, and
- Win-Trap halts exploit of buffer overflow on Windows XP;
- Stopped the execution of malicious program embedded in executable HTML pages; and
- Trapped and stopped the execution of DCOM-RPC exploit code and worm W32.Blast.Worm.

1.6 Real Stories Of Protection With Win-Trap Installed

- Win-Trap captured a SQL Slammer worm live from a vulnerable SQL server in April 2003;
- Once Win-Trap was installed on a brand-new Windows 2000 installation with the Internet connection being always on, it stopped the execution of Backdoor.IRC.Flood.E (name verified later with AV disk scan) executable planted before the installation of Win-Trap.

1.7 Pricing And Versions

Version	Protection Against Malicious Programs	Protection Against Malicious Code	Price Per CPU
Basic Version (Free for personal or research usage)	Yes	Strong against first-generation worms and hacker's code before Win-Trap	US$20.00
Professional (Enterprise users and non-critical servers)	Yes	Stronger against next generation worms and hacker's code trying to overcome Basic Version of Win-Trap.	US$40.00
Government (Government agencies and critical servers)	Yes	Strongest against any potential worms and hacker's code in the foreseeable future. So, bring them on!	TBD

Note:

- *Malicious code is defined as a piece of code to utilize vulnerabilities in applications to spread like worms such as Code Red, SQL Slammer, and to be utilized by hackers to gain control of remote computers. Malicious code requires a host application to be functional at the beginning. Malicious code might contain malicious programs within it or have the function embedded to download malicious programs.*
- *Malicious programs are self-modifying programs such as email viruses such as W32.Bugbear, W32.Klez, W32.Blast.Worm, Trojan programs, or programs embedded in self-executing HTML pages. They are executed as a standard program like other normal programs. Win-Trap does not protect against malicious scripts.*
- *Versions other than basic Win-Trap require a minimal purchase.*

1.8 Testing Is Believing

We believe that testing is believing. So, we encourage you to give us a call:

- To ask for other business info such as pricing or minimal purchase requirements.

1.9 Contact Information

Basic version is US$20.00 or Canadian $28.00 per CPU. It is free for ordinary users, students and researchers. But, you are welcome to pay for it so that we can continue developing better software for you.

Please make your money order or personal cheque payable to "**PH Security**", to the following:

PH Security
164 Equestrian Drive
Ottawa, Ontario K2M 2B9
Canada
(613) 254-5747
http://www.phsecurity.com/

Thanks a lot for your payment.

Please regularly check http://www.phsecurity.com/ first to see whether we have accepted other payment options or not.

Please contact us for site licensing and other business related inquires.

1.10 Conventions in This Document

The following is a list of typographical conventions used in this document:

Constant Width in Italic
: Is used for file and directory names, program and command names, command-line options, URLs. If it is in quotation marks, then it is a command input.

Arial Fonts in Bold
: Is used for a menu item to be selected. It is used in quotation marks.

1.11 Symbols And Acronyms Explained

<SYSTEM_DRIVE>
: Is used to indicate the drive where the operating system directories are located.

<OLD_SYSTEM_DRIVE>
: Is used to indicate the drive where the operating system was installed with the Win-Trap software package when the system is booted up by another partition or the emergency console or other means to complete the install/uninstall/repair procedure for Win-Trap. During installation and repair procedure, *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\SYSTEM32* directory has the file as

`ternel32.dll`, has a file as *`phrestore.bat`* during the uninstall procedure. These two files are indicative of where the system directory was.

<WINDOWS_DIR>

Is used to indicate the Windows directory where the operating system is located.

Note:

To find out the above three symbols, use "*`set`*" command from a command-line console, then you should see the last line of the output such as "*`windir=c:\winnt`*", where <SYSTEM_DRIVE> is *c* and <WINDOWS_DIR> is *`winnt`*.

<CDROM_DRIVE>

Is used to indicate the CDROM drive letter.

Chapter 2 Install And Uninstall Win-Trap

2.1 Install Win-Trap

Win-Trap from PH Security is released as a zipped file called *wintrap.zip*. It can be unzipped with many compression tools such as winzip. If the package is unzipped to a directory, you can start installation process by running "*setup.exe*" program from the directory. With GUI interface such as winzip program, you can start installation process by clicking on the 

icon.

Once you set the installation process, it begins with a warning screen like Figure 2-1.

> ***It warns you that the best solution against malicious programs and code is not to execute any unknown executables and apply software patches, if available, as soon as possible.***

Figure 2-1 Warning Screen

Then, it will display the license agreement in *notepad*. After you close the *notepad* windows, it will pop up such a screen as Figure 2-2.

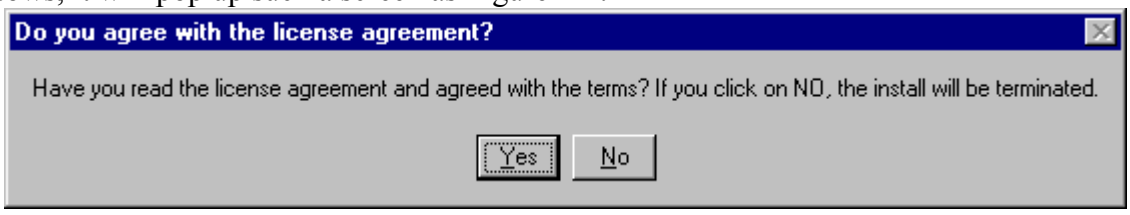


Figure 2-2 Agree to License Agreement?

Then, it will pop up the screen like Figure 2-3. Once you click on the Yes button, it will begin to install the software onto your system.

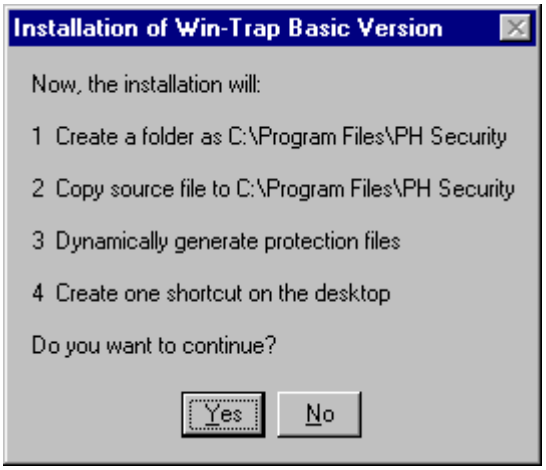


Figure 2-3 Installation Info Screen

2.1.1 Partial Installation Completion on Windows 2000, XP and 2003

Due to the fact that Microsoft implements the Windows File Protection (WFP) scheme to protect system files from being modified on Windows 2000, XP and 2003, the installation of Win-Trap core cannot be automatically completed without your manual intervention explained in 2.3.

For installation on Windows 2000, XP and 2003, it will display a warning screen as Figure 2-4.

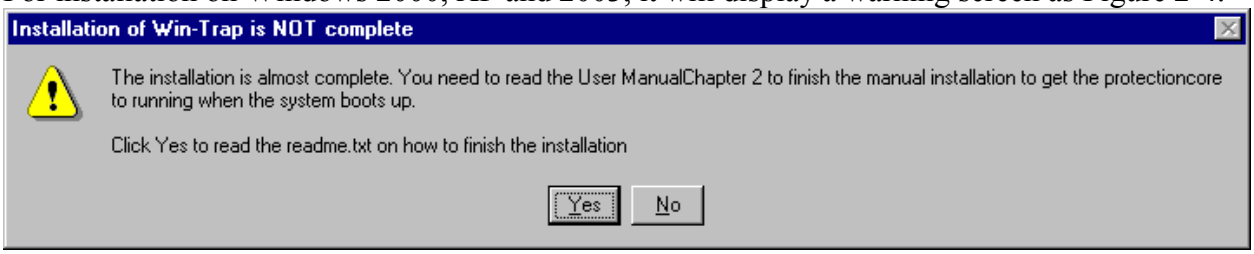


Figure 2-4 Installation Is NOT complete on Windows 2000, XP and 2003

You need to follow the next section to finish the installation process on Windows 2000, XP and 2003.

If you click on 

, it will show the next section in *notepad* program windows.

2.1.2 Complete Installation on Windows NT

However, the installation program is able to install the Win-Trap core on to Windows NT without any problem as Figure 2-5 shows. Please choose to reboot the system to make the protection mechanism of Win-Trap core take effect.

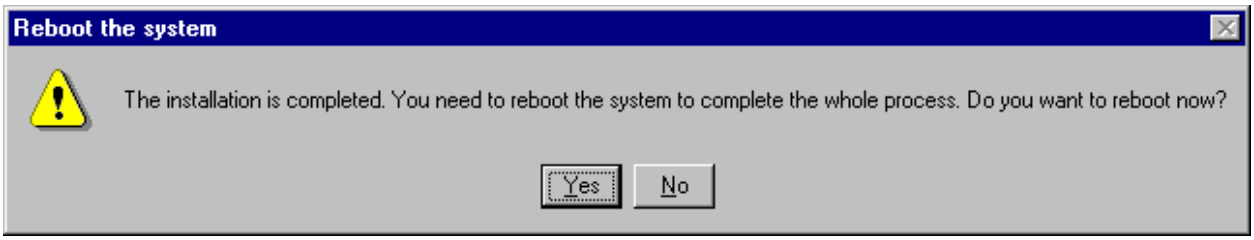


Figure 2-5 The Whole Installation Is Complete on Windows NT

2.2 Install the Recovery Console on Windows 2000, XP, 2003

If you have multiple boot partitions installed on your system, you do not have to install the Recovery Console if other operating systems can access and modify the partition where the *<OLD_SYSTEM_DRIVE>* is on. Or you have an Emergency Boot Disk/CD available to complete the tasks specified in section 2.3 and 2.4, you do not have to install the Recovery Console either.

Otherwise, please insert the Windows 2000, XP or 2003 Operating System CD to install the Recovery Console first so that you can finish the tasks specified in section 2.3 and 2.4.

To install the Recovery Console, Run "*<CDROM_DRIVE>:\i386\winnt32 /cmdcons*" and follow the instructions given.

2.3 Post-Installation on Windows 2000, XP, 2003

Microsoft implements a so-called Windows File Protection (WFP) scheme on Windows 2000, ME, XP and 2003 to prevent other applications, either normal or malicious, from modifying system files.

Unfortunately, this good intention also prevents newly invented protection mechanism such as Win-Trap from being easily integrated with the system. So, bear with us for this inconvenience to get Win-Trap technology across to protect your system from malicious code such as Code Red, SQL Slammer etc. worms and malicious programs such as W32.Bugbear and W32.Sobig etc. email viruses and Trojans.

Note:

Win-Trap product will modify some system files, especially *kernel32.dll* file under *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32*, to install the protection layer for the system. In order to keep any booting disaster (no data loss and damage at all) from happening, it is suggested that you back up the file *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32\kernel32.saved.dll* to a different filename such as *kernel32.old.dll* under *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32* directory so that the *kernel32.dll* can be restored later if needed.

2.3.1 Install With Multiple Booting Partitions

If your computer has multiple booting partitions, then you can reboot to another partition after you ran the installation procedure above.

Then, you run a command line prompt by running "*cmd.exe*" program. Please notice that the other operating system must be able to recognize the partition information for the drive where Win-Trap has been installed.

Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32*, and then run "*phinstall.bat*" batch file.

This step completes the installation procedure for Win-Trap, finally!

2.3.2 Install From the Recovery Console

If your computer does not have multiple booting partitions, but you have installed the Recovery Console option during the installation of Win-Trap product, then you boot into the Recovery Console.

Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32*, and then run "*batch phinstall.bat*" command.

This step completes the installation procedure for Win-Trap. Type "*exit*" and ENTER key to reboot the whole system.

2.3.3 Install With An Emergency Boot CD

There are many emergency boot systems available on the Internet. Some of them are free. One example is "*Emergency Boot CD*". The website is http://www.ebcd.i-am.ru/.

Note:

PH Security does not endorse its content and cannot verify its usefulness of the above website. Visit and use at your own risk.

If you are using an emergency boot CD to boot the system up, then you have to go into an interface where you can access the partition for *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32* directory. Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32* and then do the following (commands themselves might have to be modified)

- *Delete kernel32.dll*
- *Rename ternel32.dll kernel32.dll*
- *Cd dllcache*
- *copy kernel32.dll kernel32.saved.dll*
- *del kernel32.dll*

This completes the installation procedure for Win-Trap. Reboot the system.

2.4 Uninstall Win-Trap

To uninstall Win-Trap, click on the 

button shown in 3.1. It will ask whether you want to un-install Win-Trap or not as Figure 2-6.

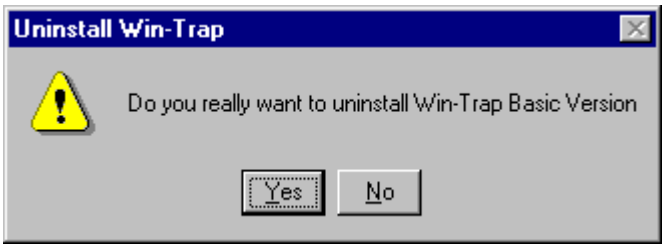


Figure 2-6 Want to uninstall?

On Windows NT, the un-installation process is complete once you reboot the system after the following prompt.

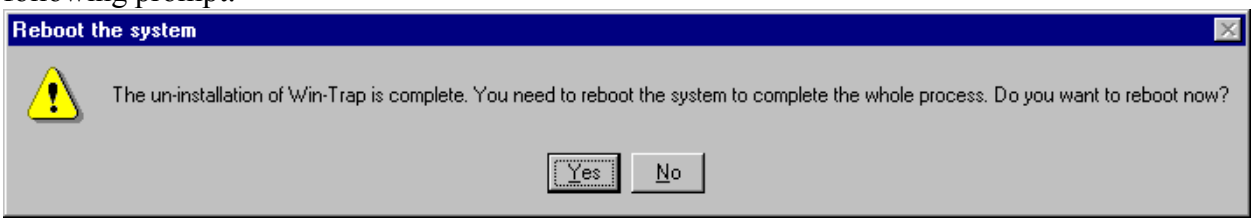


Figure 2-7 Un-installation On Windows NT

On Windows 2000, XP and 2003, it requires you to manually intervene to finish the un-installation process. Please follow the instructions in the subsection 2.4.1, 2.4.2 or 2.4.3.

Figure 2-8 Further Action Needed to Uninstall on Windows 2000, XP and 2003

Please notice the log file *<SYSTEM_DRIVE>:\PHReport.log* generated will not be deleted while the configuration file existing as *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\phsecurity.ini* will be deleted.

2.4.1 Uninstall With Multiple Booting Partitions

If your computer has multiple booting partitions, then you can reboot to another partition after you ran the uninstall procedure above.

Then, you run a command line prompt by running "*cmd.exe*" program. Please notice that the other operating system must be able to recognize the partition information for the drive where Win-Trap is to be removed completely.

Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32*, and then run "*phrestore.bat*" batch file.

This step completes the uninstall procedure for Win-Trap.

2.4.2 Uninstall From the Recovery Console

If your computer does not have multiple booting partitions, but you have installed the Recovery Console option during the installation of Win-Trap product, then you boot into the Recovery Console.

Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32*, and then run "*batch phrestore.bat*" command.

This step completes the uninstall procedure for Win-Trap. Type "*exit*" and ENTER key to reboot the whole system.

2.4.3 Uninstall With An Emergency Boot CD

If you are using an emergency boot CD to boot the system up, then you have to go into an interface where you can access the partition for *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32* directory. Then, you change the current directory to *<OLD_SYSTEM_DRIVE>:\<WINDOWS_DIR>\System32* and then do the following (commands themselves might have to be modified):

- *Delete kernel32.dll*
- *Rename kernel32.saved.dll kernel32.dll*
- *Cd dllcache*
- *Rename kernel32.saved.dll kernel32.dll*

This completes the uninstall procedure for Win-Trap. Reboot the system.

2.5 Reinstall Win-Trap

After the Win-Trap was installed, it had been functional until you installed some files from Microsoft, especially patches related to the modification of *kernel32.dll*, then you need to reinstall Win-Trap for protection.

Please just follow the normal installation procedure as outlined at the beginning of this chapter.

2.6 Thank you for paying for Basic Version

Basic version of Win-Trap is free for personal, teaching and research use. However, you are welcome to pay for it so that we can continue developing better software for you.

If you are satisfied with Win-Trap and would like to pay us for the product, please send your money order or personal cheque in the amount of US$20.00 or Canadian $28.00 (more if you want) per CPU, made payable to "**PH Security**", and mail it to the above contact address with your email address. Please check http://www.phsecurity.com/ for other payment options as our business evolves. Thanks.

The protection mechanism of basic Win-Trap against buffer overflow exploit code is not that strong even though it is strong enough to protect against current generation of buffer overflow exploit code. If you want, please contact us for stronger version of buffer overflow protection.

For ordinary users, the basic version of Win-Trap is quite useful to prevent self-modifying malicious programs from running.

Chapter 3 Normal Operation

3.1 Execute Phconfig.exe Program

Phconfig.exe is the Graphic User Interface (GUI) of the Win-Trap software package to configure protection options and analyze a logged event so that you can take further actions such as saving the logged binary data to a file for detailed disassembly later or authorizing a shareware program downloaded from the Internet.

After the installation of the Win-Trap software package, double-click the pH Sec icon on the desktop screen to execute "*phconfig.exe*" program. Alternatively, you can launch the "*phconfig.exe*" program by following the menu item starting from the Start menu -> "**Programs**" -> "**PH Security**" -> "**Configuration**".

If the Win-Trap core is properly installed, you should see a dialog screen immediately like Figure 3-1. Notice the lines and labels in RED are added to be illustrative.

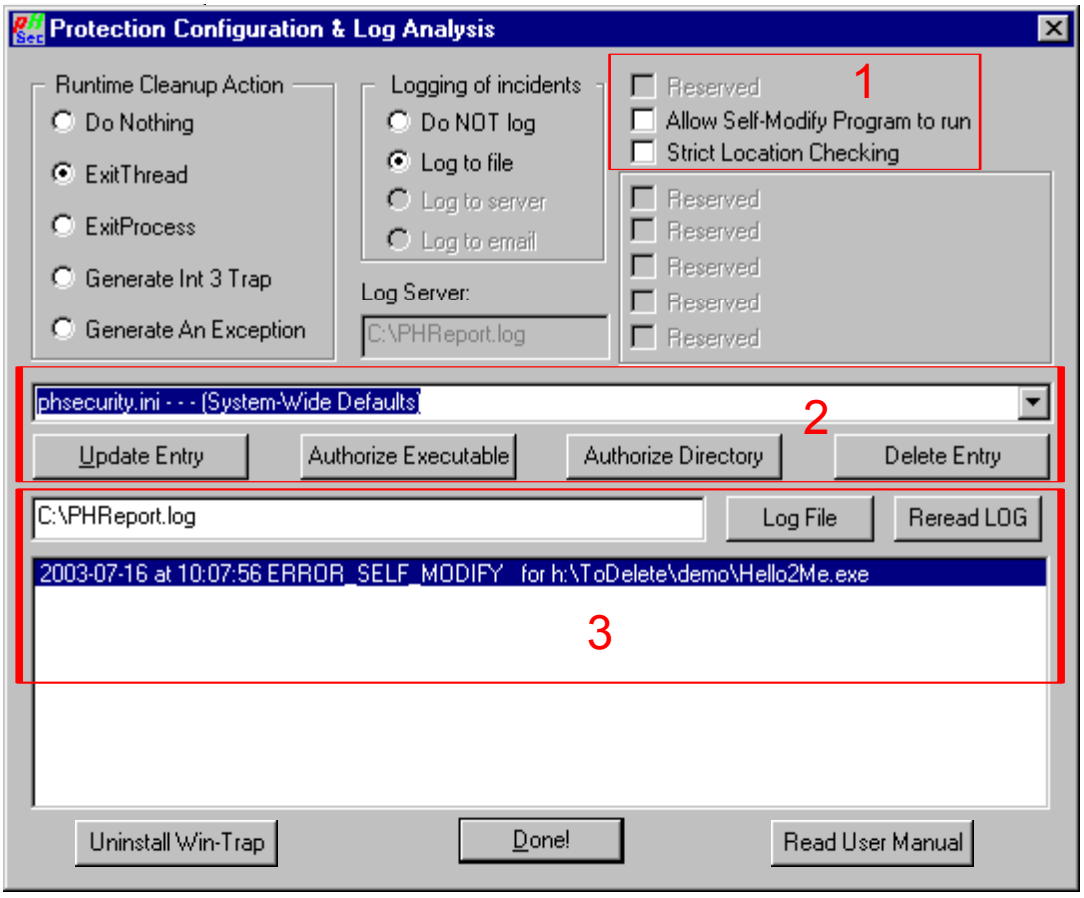


Figure 3-1 Protection Configuration And Log Analysis

If the Win-Trap is NOT properly installed or overwritten by Windows File Protection (WFP) mechanism, the warning dialog like Figure 3-2 shows up before it displays the dialog like Figure 3-1.

Figure 3-2 Warning if Win-Trap is NOT enabled

Note:

If the warning dialog like Figure 3-2 is shown, then the protection mechanisms against email viruses, malicious programs such as backdoor Trojans, and worms are NOT provided by Win-Trap. However, you can view logged events and even authorize some executable programs to run properly later. But all the changes made will NOT take effect until the core of Win-Trap software package is enabled and used by Windows operating system. This requires a reboot.

The following document assumes the Win-Trap is enabled.

3.2 Configuration For Program Initialization

When a program, either a normal application program such as Outlook or a malicious program disguised in other file extensions such as "*.pif*" and "*.scr*", Win-Trap provides protection against malicious programs during initialization with two options shown as Figure 3-3 (marked as block 1 in Figure 3-1).

Figure 3-3 Configuration Options For Program Initialization

3.2.1 Protection Against Self-Modifying Programs

The global setting "Allow Self-Modify Program to run" for "phsecurity.ini - - - (System-Wide Defaults)" (marked as block 2 in Figure 3-1) is checked off as default, then a malicious program will be terminated with a dialog as Figure 3-4 if the program is to be executed. However, normal programs will not hindered in any way.

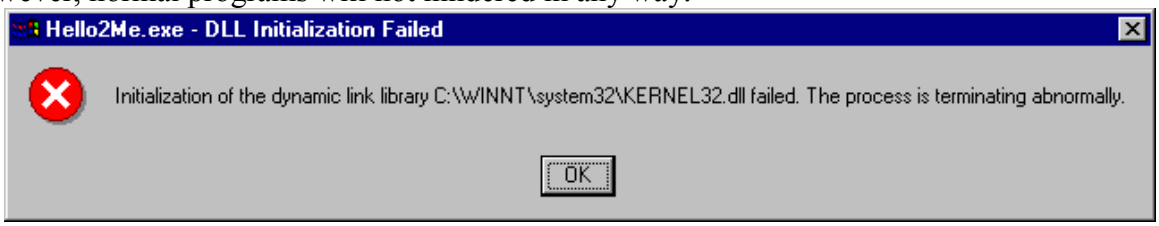
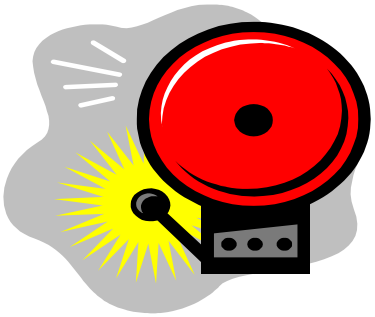


Figure 3-4 Potential Malicious Program Is Terminated During Initialization



It is very dangerous to check the box ""Allow Self-Modify Program to run" for "phsecurity.ini - - - (System-Wide Defaults)"!!! It disables the protection against new malicious programs such as email viruses and backdoor malicious Trojans. Therefore, it is strongly discouraged to check this box for global setting. ***Handle With Care!***

3.2.2 Authorize Self-Modifying Shareware

If the logged event is believed to be from a shareware, then you can double-click on the item inside the list box marked as block 3 in Figure 3-1, then you will have a dialog like Figure 3-5. Expired shareware version Win-Trap will NOT allow to bring the dialog up.

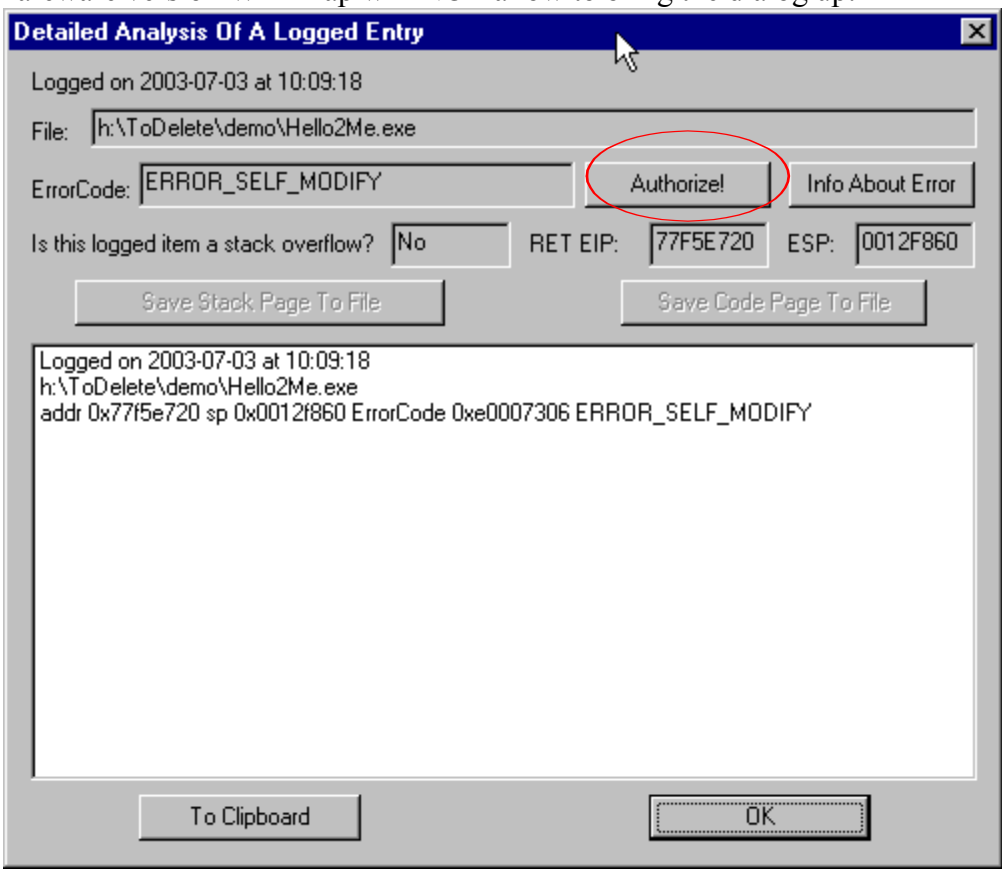


Figure 3-5 Authorize A Self-Modify Program

This dialog shows it is "ERROR_SELF_MODIFY". Other information provided is not very relevant to this type of error. You can click on Info About Error button to get more information about the error code. If you are sure, then you can authorize the logged executable to be executed next time by clicking on Authorize! button. Once you click on, it will provide explanation and warning before it ends up with a screen like Figure 3-6. Notice the block 2 is changed to the full path for the executable program.

With dialog box as Figure 3-5, you can copy the analysis to clipboard by clicking on To Clipboard button.

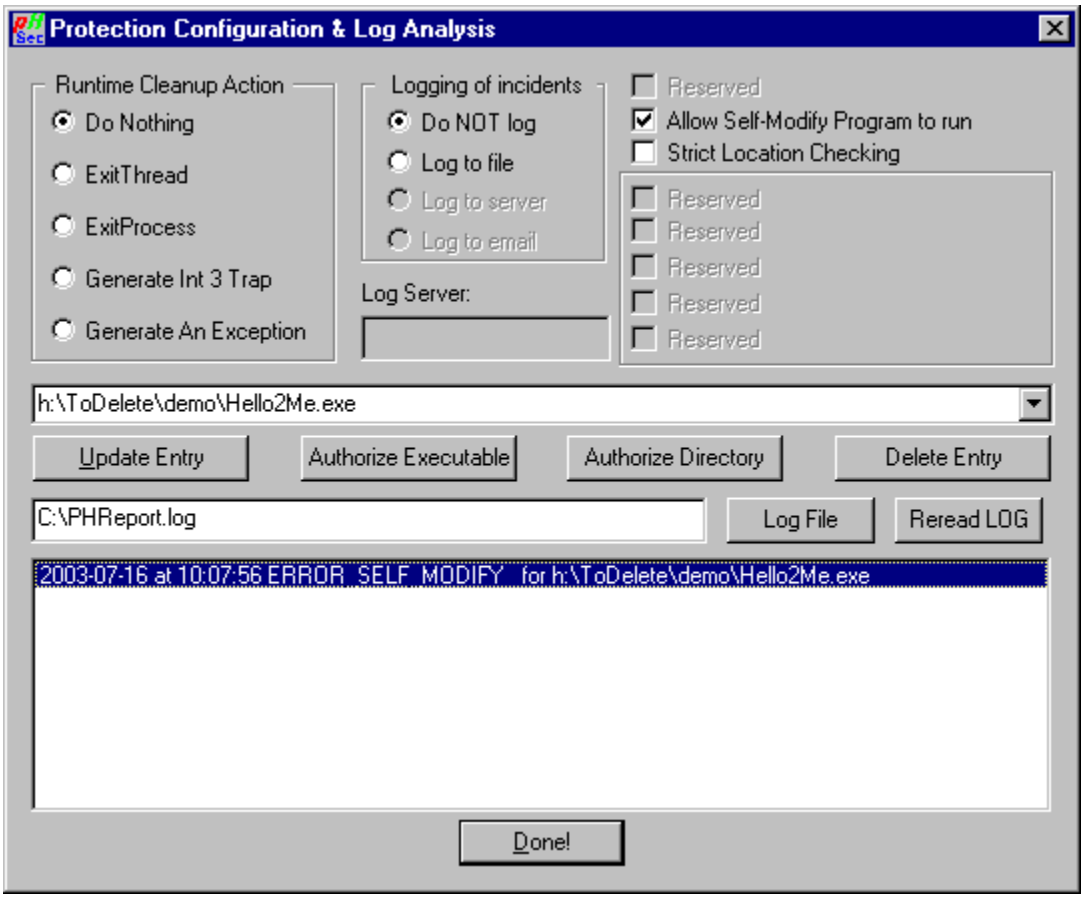


Figure 3-6 A Self-Modifying Program Is Authorized To Run

3.2.3 Very Restrictive Location Checking

The Strict Location Checking global option is very restrictive. Usually, you do not need to have this option checked to protect against execution of malicious programs. However, some malicious programs are as normal as an application program can be, they are not self-modifying programs.

With this option enabled, then Win-Trap core will check the executable file path to determine whether the executable file to be initialized is from:

- The system directory. Use the **`GetSystemDirectory`** function to get the path of this directory.
- The Windows directory. Use the **`GetWindowsDirectory`** function to get the path of this directory.
- The directories that are listed in the PATH environment variable.
- The authorized directory managed by Win-Trap GUI `phconfig.exe` program;
- The authorized executable file.

With this option enabled, a dialog screen like Figure 3-4 will pop up quite often, especially for basic version Win-Trap, which does not include programs to scan all the drives for existing normal executables to be included in the authorization list.

With this option on, you will have lots of logged events with error code as "ERROR_AUTHORIZATION" as Figure 3-7 shows. If necessary, you can authorize the execution of the logged executable file by following the steps listed section 3.2.2.

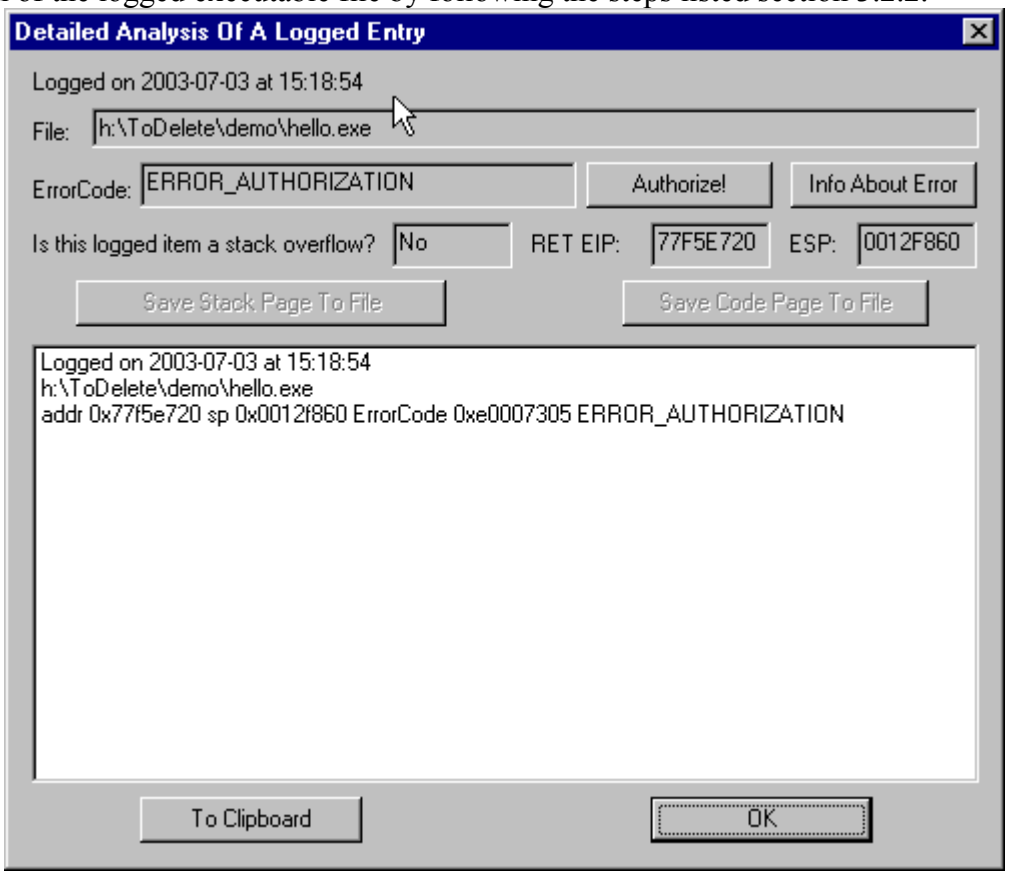


Figure 3-7 An Executable Was Executed From Unauthorized Path

3.3 Run-Time Protection Against BOF Exploitation

Exploitation of buffer overflow (BOF) vulnerabilities is quite common and sometimes very devastating such as Code Red and SQL Slammer worms. Malicious worms caused tremendous financial damages valued in billions of US dollars. Hacker's exploit code, quite often exploiting BOF vulnerabilities, is used to do virtual break-in and steal sensitive data ranging from credit card to other personal ID information.

Win-Trap provides run-time protection for normal application and server programs alike against exploitation of buffer overflow vulnerabilities, if any. When Win-Trap detects a BOF exploitation event, it will act according to one of the following options shown Figure 3-8.

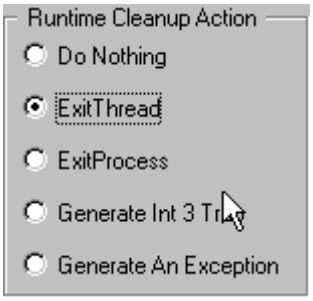


Figure 3-8 Cleanup Options When A Worm Is Captured

The  option is obviously a disabling option for Win-Trap. This option is used for some shareware programs, which are self-modifying executable files themselves or call some self-modifying dynamic link libraries (DLL). For a normal application or server program, this option should NOT be chosen.

The  option is used by Win-Trap to exit the thread which triggered its trapping. For a single-threaded process, it basically exits the whole process. For a server process having multiple threads, this option will terminate only the thread triggered the trapping. However, some functionalities of the whole process might not behave completely as originally designed. This is the default global setting to be less disruptive when an event happens.

The  option is used by Win-Trap to exit the whole process when an exploitation incident of buffer overflow vulnerability happens.

The  option is used by Win-Trap to generate a single-step exception when an exploitation incident of buffer overflow vulnerability happens. If chosen, it might display a dialog as Figure 3-9 or as Figure 3-11 or does not display anything at all, depending on the application and the operating system.

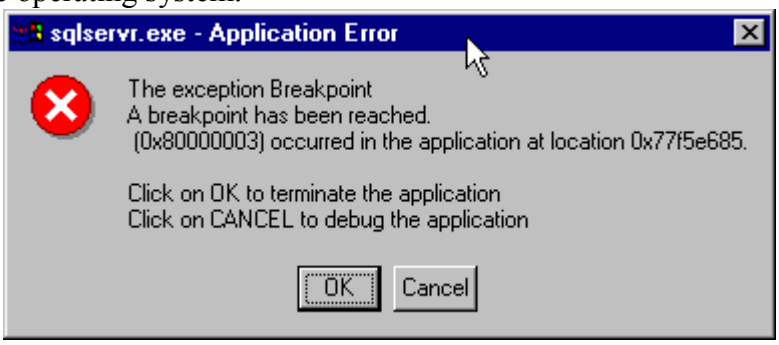


Figure 3-9 Breakpoint Exception Generated

The  option is used by Win-Trap to generate a software exception 0xE0007384 when an exploitation incident of buffer overflow vulnerability happens. If chosen, it might display a dialog as Figure 3-10 or as Figure 3-11 or does not display anything at all, depending on the application and the operating system.

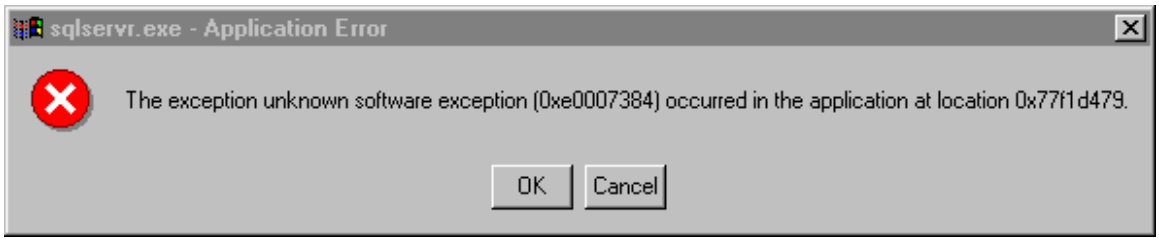


Figure 3-10 Software Exception 0xE0007384 Generated

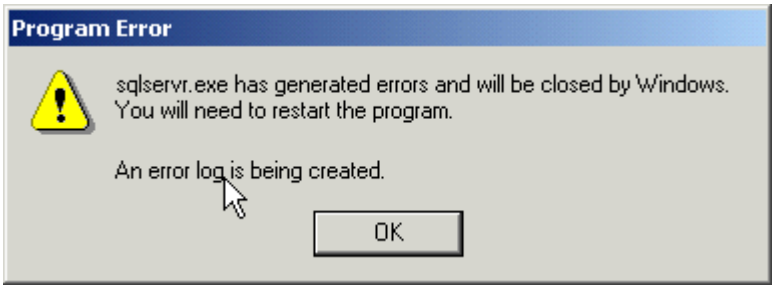


Figure 3-11 Exception Error On Windows 2000

Notes:

Some malicious code hooks up with Frame-Based Structured Exception Handling (SEH) and might be able to utilize the last two options listed above. However, these two options are provided here to have some graphic and debugging functionalities. For a normal operation system, choose either ExitThread or ExitProcess to clean up when an exploitation incident is trapped within a vulnerability process.

3.4 Event Logging

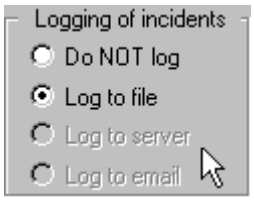


Figure 3-12 Logging Options

For now, there is only one of two logging options. If Log to file is chosen, then it will log an event, either an initialization failure of a self-modifying program or an exploitation of buffer overflow, to the file `<SYSTEM_DRIVE>:\PHReport.log`.

During the initialization of an application, Win-Trap will log the error, either `ERROR_SELF_MODIFY` or `ERROR_AUTHORIZATION` only once. During normal run-time of an application, Win-Trap will only log three incidents per process at most to prevent the log file from getting too big if Do Nothing or ExitThread run-time cleanup action option is chosen.

3.5 Normal Application With Self-Modifying DLL

A few application executables themselves are not self-modifying programs, but they use self-modifying dynamic link libraries (DLL). Then Win-Trap will detect this and treat it as an exploitation event. If an applications runs properly before the installation of Win-Trap and fails to work after the installation of Win-Trap, then you can choose Do Nothing and turn off the logging to disable the Win-Trap protection for that particular application by using `phconfig.exe` program. The error code for it is ERROR_WRITEABLE.

Self-modification mechanism is vigorously exploited by computer virus writers to prevent the anti-virus researchers from learning and understanding viruses' internals, and used by some shareware, trial-ware writers to hide their registration key verification mechanism.

Please notice the majority of application programs are not self-modifying and do not call self-modifying dynamic link libraries (DLL).

3.6 Analysis Of Buffer Overflow Exploitation

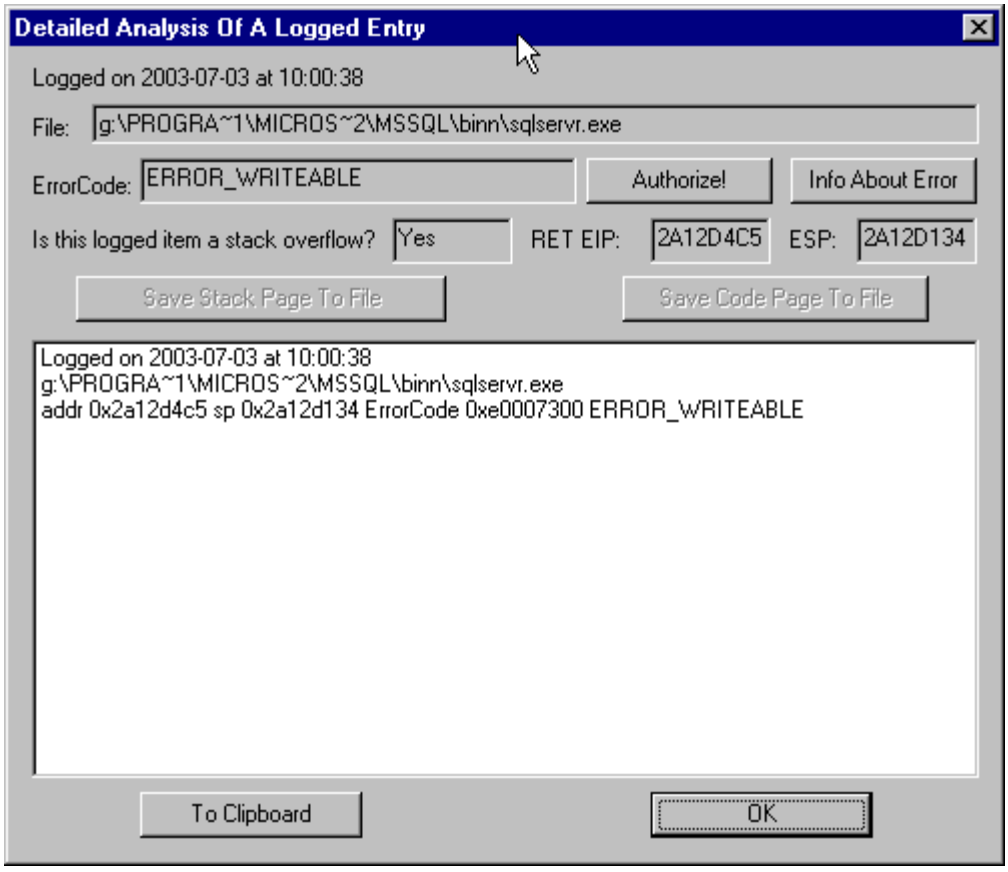


Figure 3-13 Analysis of an exploitation of buffer overflow incident

For a normal application, no error log will be generated except when there is a malicious exploitation of buffer overflow. For example, Figure 3-13 illustrates a logged event for the exploitation of SQL Slammer buffer overflow vulnerability.

The "Detailed Analysis Of A Logged Entry" dialog box enables you to view the EIP and ESP CPU registers and informs you whether this is a stack-based overflow or not when this log was generated. If you believe this log was generated by a self-modifying DLL as explained in section 3.5, then you can authorize the executable by clicking Authorize! button.

With this dialog box, you can save the binary data on the offending code page to a file to reverse, and/or save the stack page to analyze the stack trace. The EIP or ESP has to be turned into a page offset for investigation. It is calculated as:

Offset = EIP & (PAGE_SIZE – 1) = EIP & 0xFFF on x86

Then, you can use this offset value to index into the page file(s) saved for clues to what happened.

3.7 Log File

The Win-Trap core saves the logged events to a file *<SYSTEM_DRIVE>:\PHReport.log*. However, you can choose a different log file to analyze by clicking on Log File button. If the default log is used and *Phconfig.exe* program would not automatically read an updated version log file written by the Win-Trap core. You have to click on Reread LOG button to reread the log file.

Figure 3-14 Log File Location

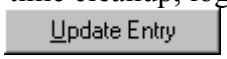
3.8 Authorization

Figure 3-15 Authorization Options

Figure 3-13 illustrates the authorization options for the global protection when there are no specific protection options set for a program or the directory where the program is located. You can click on the  to select one existing item from the drop-down list to update.

3.8.1 Authorize Programs Within A Directory

To authorize the execution of programs within a special directory, click 

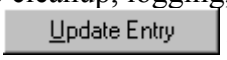
and select any file within a particular directory and then the directory will be selected in the selection line, you can choose options for the run-time cleanup, logging, and self-modification code. Once you finish your selection, you can click 

to update the directory protection option entry.

All the programs within the directory will be protected according to the options selected if there is no authorization entry in the authorization list for a program within the directory. However, programs within subdirectories of the directory will NOT be protected according to the options selected for the directory.

3.8.2 Authorize One Particular Program

To authorize one particular program, click 

to select the full path for that program. Then, you choose options for the run-time cleanup, logging, and self-modification code. Once you finish your selection, you can click 

to update the directory protection option entry.

The authorization options for this particular program will apply only to the program for protection or lack of it.

3.8.3 Update And Delete An Entry

You click on the  to select one existing item from the drop-down list to update. To delete an item entry, select an existing item and then click 

.

Chapter 4 Frequently Answered Questions

1. One program used to work and stops working after installation of Win-Trap, what happened?

   - First, have you checked the event log generated by Win-Trap core with the *phconfig.exe* GUI program? If there is an event corresponding to the program, try to authorize it to run since it could be a self-modifying program or a program calling self-modifying DLL.

     Unfortunately, some programs do not behave well, especially shareware. So far, we noticed only one shareware program with encrypted DLL do not follow the normal DLL standards and therefore it refuses to work properly with Win-Trap core without any error messages generated from either the operating system or the Win-Trap.

2. Should I back up the file *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\system32\kernel32.saved.dll?*

   - Yes, it is a good practice to save the *kernel32.saved.dll* to *<SYSTEM_DRIVE>:\<WINDOWS_DIR>\system32* just in case. If something happens, then you can restore the *kernel32.dll* from the saved file after it is booted up from the Recovery Console or the Emergency Boot Disk.

     kernel32.saved.dll is generated during the installation/repair of Win-Trap software package and is used to restore back as *kernel32.dll* when the Win-Trap software is uninstalled.

3. How protective is the Win-Trap?

   - The primary functionality of Win-Trap software package is against malicious exploitation of buffer overflow vulnerabilities existing in applications, either an office productivity application or a server application. The basic version of Win-Trap is very effective against today's worms and hacker's code already. However, it is realized that the computer worms and hacker's code are evolving as well to overcome the protection layers you put up. Therefore, the professional and government versions of Win-Trap software package provide protection against next generation of exploit code trying to overcome the basic version of Win-Trap. With the deployment of highest protection version of Win-Trap, it is possible in theory to overcome the protection provided by Win-Trap, but it becomes too practically difficult or impossible to write an exploit code even if there are some buffer overflow vulnerabilities present in application programs.

     The secondary functionality of Win-Trap software package is against malicious email attachments disguised in different file extensions. Malicious email attachments have many forms such as VBScript, self-executing HTML pages, and executable programs in Portable Executable File format. Win-Trap is effective against self-modifying executable programs in Portable Executable File format and does not protect against scripts.

4. The installation of Win-Trap failed, what happened?

   - There could be many reasons for the failure of installation. One possible reason is that *kernel32.dll* has been modified by other programs. You are welcome to send the <SYSTEM_DRIVE>:\<WINDOWS_DIR>\system32\kernel32.dll to us to investigate. Please notice this file does not contain any sensitivity information except it is a highly customized version, specified by other vendors' documents.

5. I upgraded my computer from Windows NT to Windows 2003, should I get a new copy of Win-Trap?

   - No, you do not have to unless you want to upgrade to a stronger version of Win-Trap. Win-Trap installation uses dynamic code generation and will work with new OS as well.

6. I applied a patch from Microsoft to my existing Windows. Should I get a new copy of Win-Trap?

   - No, you do not have to unless you want to upgrade to a stronger version of Win-Trap. Win-Trap installation uses dynamic code generation and will work with the new patch as well. It is possible that you have to do uninstall/install process if the new patch from Microsoft replaces the old *kernel32.dll* with a new one.

7. Win-Trap sounds great. But does it prevent the buffer overflow from happening?

   - No, Win-Trap does not prevent the buffer overflow from happening at all. It is the programmer's responsibility to make sure that its code does not have any buffer overflow vulnerability. It is the vendor's responsibility to provide and notify its customers that there are patches available to fix buffer overflow vulnerabilities if any. Win-Trap provides the protection after malicious code is used to intentionally overflow a buffer vulnerability and begins to exploit it. Win-Trap does not prevent the single-shot Denial of Service (DoS) due to the exploitation of one buffer overflow vulnerability. But, Win-Trap does prevent this exploitation from going further and therefore reduce the potential financial or security risks associated with malicious exploitation of buffer overflow vulnerabilities as Code Red, SQL Slammer and others have caused. Depending on the version of Win-Trap you use, the protection strength varies.

8. Win-Trap sounds fabulous. But does it prevent all the executables in PE format from running?

   - No. It does not prevent malicious programs in normal PE format from running. With "Strict Location Checking" global option on, it is possible to halt a new malicious program in normal PE format from running. But, the malicious program can still slip through even though it is unlikely. If financially sustainable, PH Security might be able to develop a product to differentiate a malicious program in normal PE format from a

normal program in normal PE format. So, please pay us to develop protection tools for you if you are using Win-Trap basic version.

9. If there is a problem with the system after the installation of Win-Trap, what should I do?

   - First of all, try to narrow it down to the possible cause of the problem. If you are quite sure it is the Win-Trap causing the problem, you can uninstall the Win-Trap. If Win-Trap is uninstalled, and the problem persists, then it is very unlikely that the Win-Trap is the cause of the problem.

   - We tried very hard to test Win-Trap. However, we cannot test it out against all the scenarios and you are welcome to report the problems to us.