



# Crystal Anti-Exploit Protection User's Guide

# Product Overview

## **What is Crystal AEP?**

Crystal Anti-Exploit Protection is software designed to help decrease the risk posed to users of Microsoft Windows by software vulnerability exploits. Exploits are malicious pieces of code which allow a hacker or author of malicious software (malware) to run their code on your system.

The exploits which Crystal aims to block typically work by triggering a coding or design weakness within a popular Internet-accessible software package (such as a web browser, email client or word processing package), tricking the software into running code which it was not designed to run.

Crystal AEP works by inserting itself within an at-risk program and altering the way the software program behaves by making the behaviour of the program less predictable for an attacker and by inserting important security checks at certain pivotal points used by many software exploits.

## **How does Crystal AEP differ from Antivirus software?**

Crystal AEP does not contain any signatures which it could use to recognise malicious code and does not attempt – in general – to use heuristics to determine whether software is safe to run.

Whereas an antivirus package generally matches all opened or executed files against a set of virus signatures (which are continually updated) and is generally therefore quite effective at catching well known threats, it often offers little to no protection against novel threats for which signature updates have not yet been created.

Antivirus software also operates by attempting to address the symptom rather than the cause of malicious software: Antivirus rarely attempts to block the exploitation of the software vulnerability which lead to the installation of malicious software, and instead blocks the malicious software itself. Therefore it is generally a simple matter for an author of malicious software to modify the malicious software to evade detection rather than requiring them to find a new flaw through which to load the malicious code on a victim's system.

Crystal AEP is often able to block the root cause of infection by malicious software by removing the malware author's ability to run malicious code in the first place, and unlike antivirus software does not attempt to detect malicious software based on signatures or by using any comparable method.

Crystal AEP is not designed to replace antivirus software and does not offer any protection against user error – if users are fooled into downloading malicious programs sent to them by email or otherwise then Crystal will do little to help. Crystal is designed to prevent users from being infected passively through “drive-by download” exploit techniques in which a user, visiting a web page or opening an email, is automatically exploited otherwise without any user error.

Crystal also provides content filtering capabilities to Internet Explorer, allowing third parties to extend the product to add security checks and improvements as they are needed, especially to help mitigate certain threats against users of online banking or other high-value web services.

# Supported Products

## **Which operating systems and products does Crystal support?**

Crystal is designed to support Windows XP, Vista and 7, both 32-bit and 64-bit versions. Crystal AEP does not protect 64-bit applications and will, on 64-bit Windows, protect only 32-bit processes. Crystal may work on server environments such as Windows 2000, 2003 and 2008 but has not been tested on these OS.

Crystal only protects 32-bit software running on 64-bit Windows. 32-bit software constitutes the vast majority of exploited software of Windows platforms and is the least well protected by the operating system in all versions of Windows.

## **Which web browsers does Crystal provide content filtering protection to?**

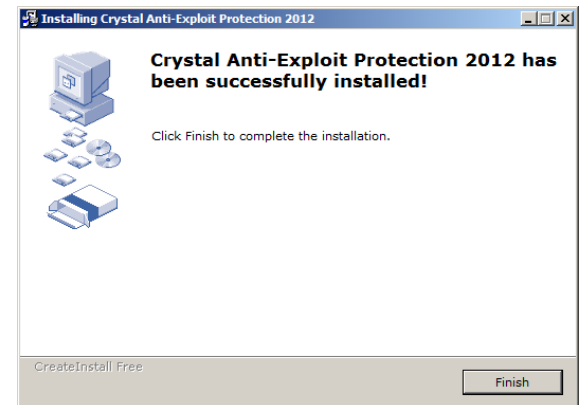
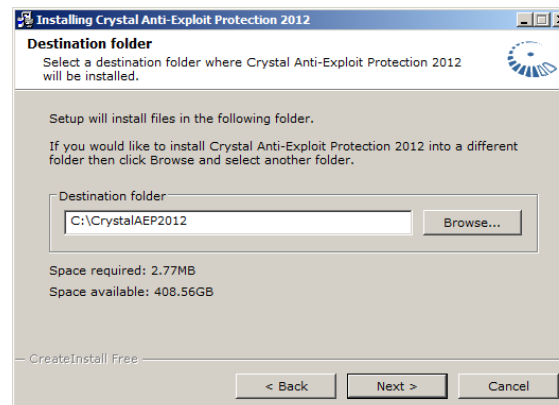
Crystal only provides content filtering protection to Internet Explorer 6 – 9 (IE9 is the latest version at the time of writing). Crystal provides anti-exploit protection to all web browsers, however the content filtering protection method can only be applied to Internet Explorer at present.

If you do not use Internet Explorer it is still worthwhile using Crystal as content filtering is at present a secondary feature of the product and is not relied on heavily to block malicious software.

# Installation

## How do I Install Crystal AEP?

Installing Crystal AEP is straightforward and should be able to be performed readily using the installer, which appears as follows:



Upon installation the Crystal AEP icon should appear in the top right corner of the screen. If the icon is blue then Crystal is installed properly and is protecting your system. If the icon is red then the system requires a restart. If after a restart the icon is still red then the installation did not complete properly and the software should be removed and then reinstalled.

A blue icon indicates that Crystal AEP is enabled.

A red icon indicates that Crystal AEP is disabled.



### **Does Crystal require any pre-existing software to run?**

Crystal requires the .NET Framework 2.0 to run the user interface (an important part of the software!). Crystal does not require the user interface, but it is recommended to configure the software.

As most systems have the .NET Framework installed there are usually no steps required to ensure that Crystal will run. If, during or after installation the install program alerts you that .NET 2.0 was not found, or if the Crystal user interface does not run when you run the program shortcut, please download the .NET Framework 2.0 from:

<http://www.microsoft.com/download/en/details.aspx?id=19>

### **Can Crystal be run under a non-administrative user account?**

No, the Crystal user interface requires access to aspects of the Windows registry which are not accessible to non-administrative users. Crystal does however work well with UAC and should only prompt for elevation when it is required.

### **Does Crystal work on multi-user PCs?**

Crystal should work fine on multi-user PCs if only one user is logged in at a time. Scenarios in which two or more users are running Crystal at once on the same machine (for example, two users remotely logged in to a PC) are not currently supported.

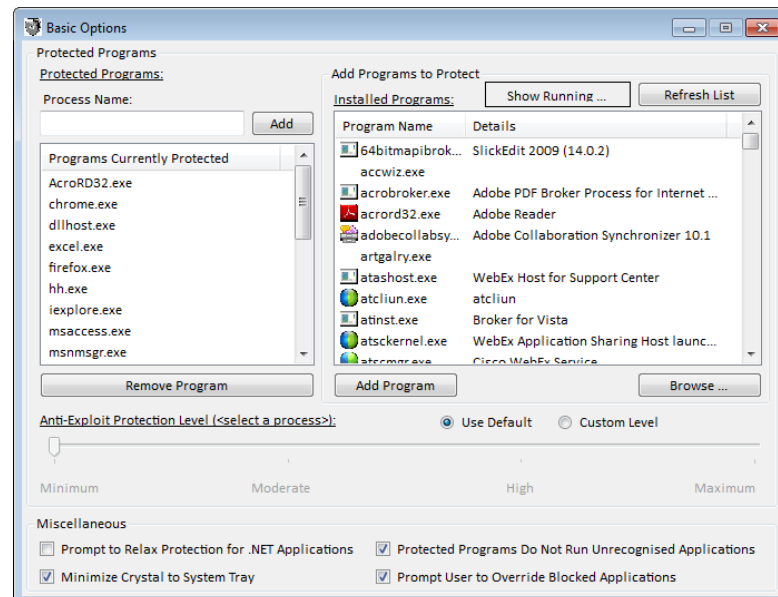
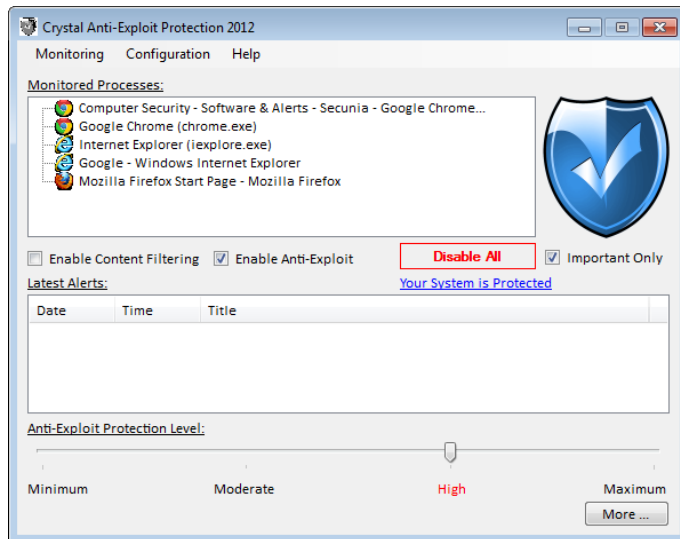
# Adding Programs to Protect

## How do I add a program to Crystal for protection?

Once Crystal is installed and is running with a blue icon indicating that installation succeeded it is necessary to choose software programs which you desire Crystal to protect. By default Crystal provides protection to:

Internet Explorer, MSN Messenger, PowerPoint, Outlook, Word, Excel, VLC Media Player, Windows Media Player, Adobe Acrobat Reader, Chrome, Firefox and RealPlayer.

To add software to protect, first click the blue Crystal desktop icon to bring up the main software view, and then navigate to Configuration, Basic Options:



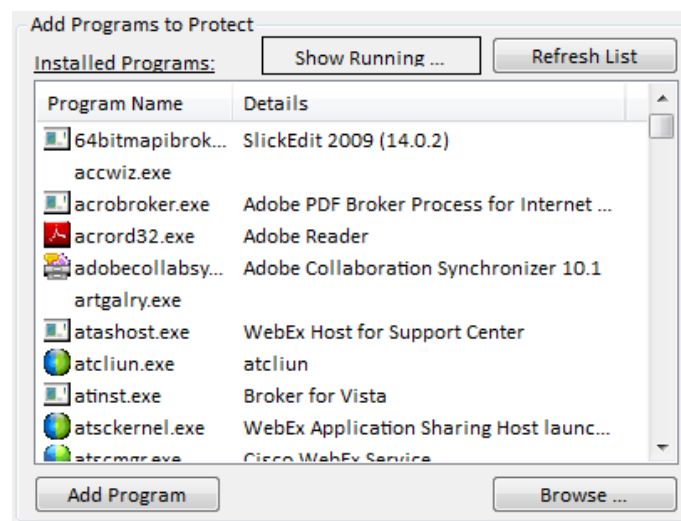
A list of programs which Crystal detects as installed is displayed on the right hand side of the screen:

To add new programs, scroll through the list of installed programs and select the program you wish to protect from the list. Then click the Add Program button at the bottom of the list.

Alternatively, if you cannot find the program you wish to protect in the list of installed programs, please start the program you wish to protect so that it is currently running, and select the depressed "Show Running ..." button at the top of the installed programs list.

The list should be repopulated with a list of currently running software from which you should be able to see the program which you intend to protect.

Select the Add Program button at the bottom of the form to add the program. It will be protected at the default level of protection configured in the main interface of Crystal (the image is from the main form, not the Basic Options form):





It is possible to configure the protection levels for software individually so that, rather than simply using the same protection level for all software, individual programs can be set to higher or lower defence thresholds.

To configure the protection levels for a program, within the Basic Options form first select – from the list of protected programs at the leftmost side of the form – the program you wish to alter the protection level for. At the bottom of the Basic Options form you should see a set of options (“Use Default” and “Custom Level”), and a slider:



Select “Custom Level” and then move the slider to configure the level of protection you feel is appropriate for the program you are protecting:



If you do not set any protection level and instead leave the software to use the default protection level then the protection level applied is that which is configured using the corresponding slide in the main user interface

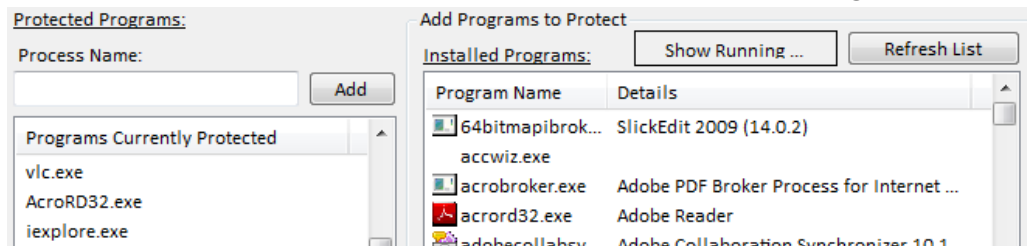
## How do I match a process name with a program name on the Protected Programs list?

In the Basic Options form you may have noticed that protected programs listed at the leftmost side of the form are listed by their short name and not the full name of the software package, for example Microsoft Office Word is listed by its executable name "winword.exe".

This is because Crystal is agnostic to the install path of software it seeks to protect and is able to protect software if it is copied or moved to a different path on the system. In most cases the name of the software program being protected can be inferred from the short name, for example:

vlc.exe = VLC Media Player  
iexplore.exe = Internet Explorer  
firefox.exe = Firefox  
wmplayer.exe = Windows Media Player

In some cases however the short name can be confusing. In these cases please take steps similar to those advised when identifying software to add for protection, and match the short name with the program name in the Installed or Running Programs list, for example, with AcroRd32.exe you can see the associated program is Adobe Reader:



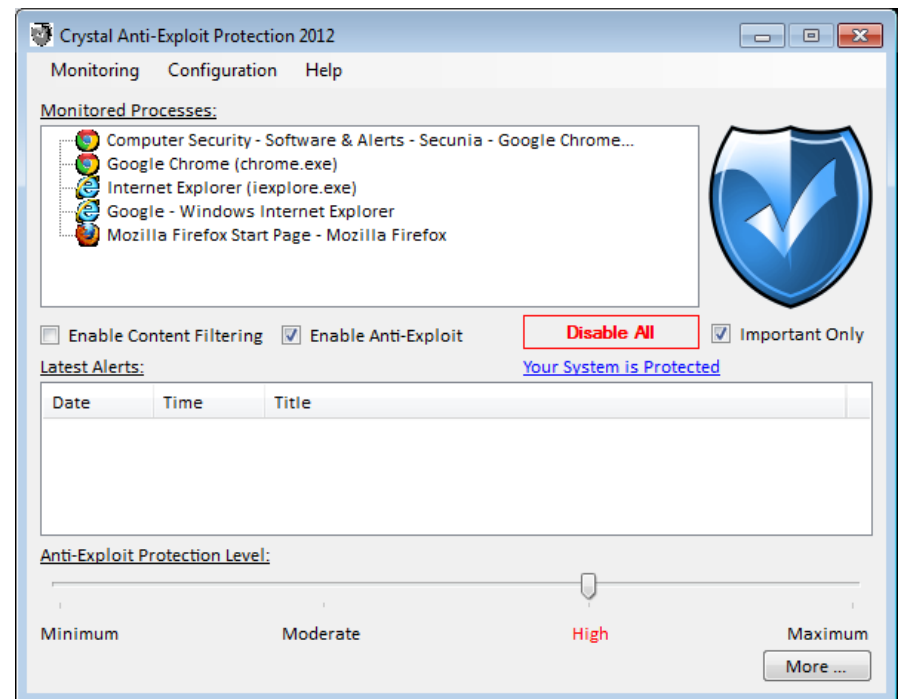
# The Main Interface

The main interface of Crystal AEP indicates which running programs are being protected, what protection level is configured for protected programs (with the exception of those configured individually), and provides information on threats which the software blocks in the form of real-time alerts:

Under the Monitored Processes list, instances of programs which are currently protected are listed. If you have added a program to protect and it is not showing up under this list it may be necessary to restart the program you are trying to protect.

Underneath this view are settings to enable or disable features of the product such as Content Filtering and Anti-Exploit features. To disable the entire product just toggle the Disable All button.

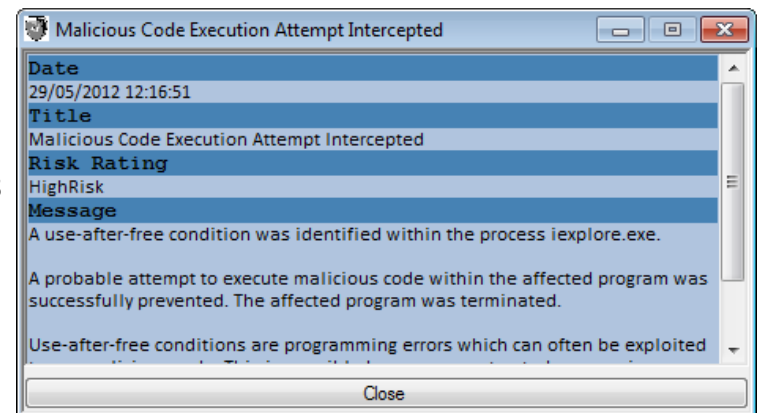
Whenever a possible security flaw is blocked by Crystal an entry is added under the Latest Alerts view. High risk issues are popped up automatically to alert the user.



To view an alert simply double click the alert title, and the following window appears:

Alerts display some detailed information on the threat which was blocked by Crystal.

As code execution exploit threats generally put the affected program in a dangerous state Crystal immediately terminates any program which it detects as being in a dangerous state. This makes it difficult for Crystal to record very specific information relating to the nature of certain types of threats, however the fact that such threats were blocked successfully is always recorded.



The Anti-Exploit Protection Level slider is a convenient way for users to configure the level of protection applied to their programs. The protection levels can be summarised roughly as follows:

Minimum – Provides only the basic protection, none of which should be invasive or disrupt delicate programs. This mode provides a backstop against some classes of threat and is surprisingly effective considering the limited features that it enables, but is not recommended for most programs as little is done to disrupt exploit attempts.

Moderate – A good improvement on Minimum, this mode aims to provide a compromise between reliability and security, erring on the side of reliability. This mode is recommended for applications which do not cope

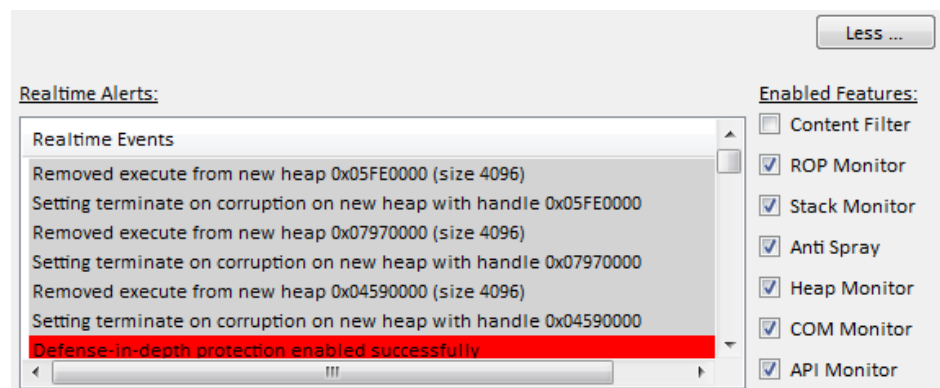
well with the High mode of protection, but is otherwise not recommended.

High – Provides an equal balance between reliability and security. Most of the particularly effective anti-exploit techniques are enabled when using this mode. This is the recommended mode for most users.

Maximum – This mode provides the highest level of protection which Crystal affords, enabling nearly all of the protection features the product can offer. Occasionally Maximum protection provides too locked-down an environment for flexible programs to operate under, and is therefore not recommended above High for most users. Maximum can be enabled for systems for which security is absolutely paramount above software reliability.

Expanding the main interface by clicking the “More ...” button, users can view real-time information on protected programs. This information allows a user to know that Crystal is running properly and may occasionally be useful in diagnosing problems or compatibility issues when attempting to protect programs.

To view real-time alerts, ensure that a program is selected under Monitored Processes at the top of the main interface.



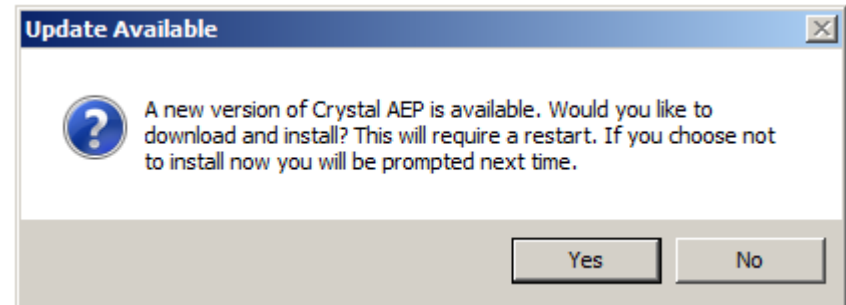
# Updating Crystal

When the Crystal AEP interface runs at Windows start-up or when launched through the desktop or start menu shortcut, the product automatically checks for updates. To check for updates manually, please navigate to the menu item Configuration, Update in the main interface.

If an update is available the user will be prompted to update and – if the prompt is accepted – the update process will be performed automatically.

If for some reason the product does not report updates for a long period of time or the update process fails, please attempt to update manually by downloading a new installer from:

<http://www.crystalaep.com/download.htm>



# Expert Options

Crystal provides the ability for expert users to configure precisely which protection methods are enabled system-wide or for a specific application, and allows users to enable/disable content filters.

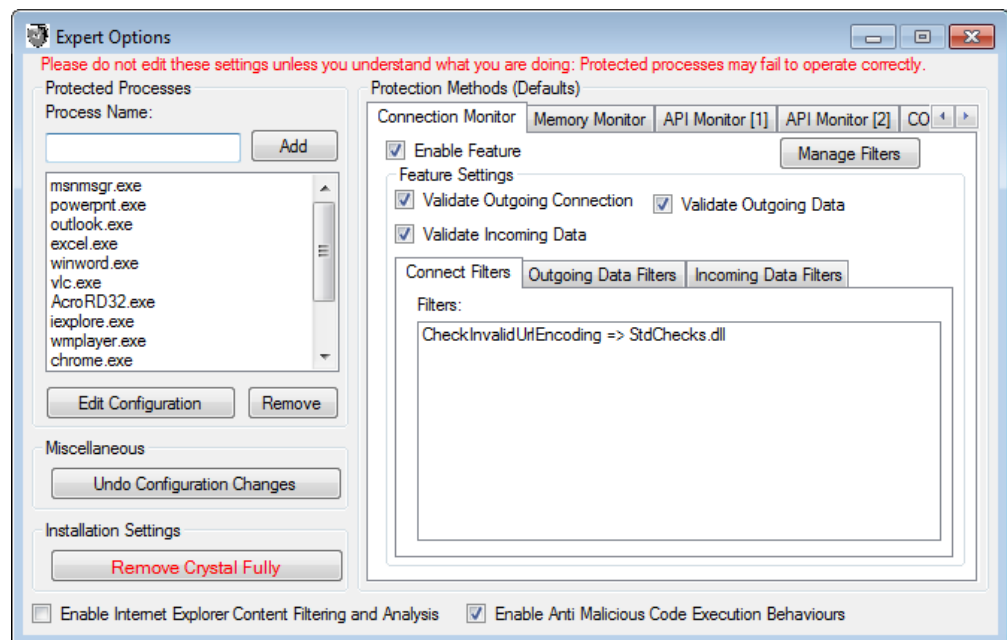
Please note that Expert Options should not be tampered with unless a user understands what they are doing, otherwise protected programs may fail to run properly.

The Expert Options form is available under the menu Configuration, Expert Options. The form appears as follows:

The Protected Processes list mirrors the Protected Programs list in the Basic Options form.

By default alterations made in the Expert Options form affects all programs.

To edit configuration for a specific program it is necessary to select the process name from the list and click the Edit Configuration button.



It will be evident when program specific options rather than system-wide options are being edited – the option text will appear red (as follows):

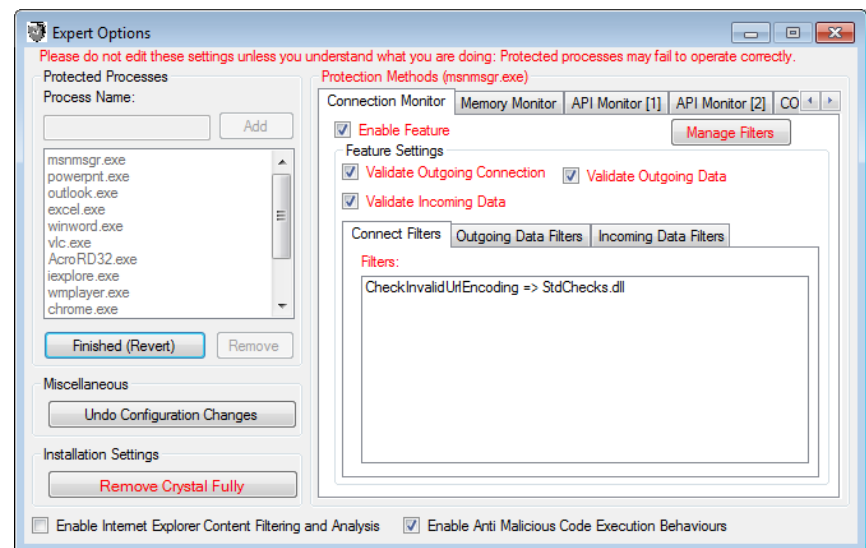
There are four main categories of settings which can Be edited under the Expert Options form. These categories are:

- Connection Monitor
- Memory Monitor
- API Monitor (two pages)
- COM/ActiveX Monitor

These categories provide the following possibilities:

### Connection Monitor

The Connection Monitor category contains settings which apply to content filtering. Within this view of the Expert Options form it is possible to enable or disable the content filtering feature, or to do the same to individual categories of filter, such as:





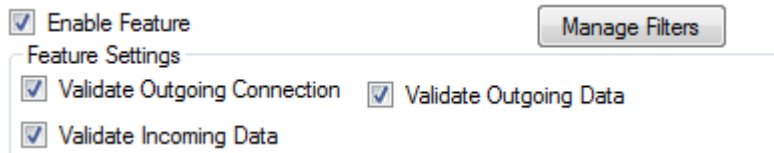
*Connect Filters* ("Validate Outgoing Connections") – filters which inspect attempts to connect to a remote website and determine whether the connection should be allowed. Such filters may choose to block entire websites (for example, fake online banking sites or sites with an IP-address rather than a hostname).

*Outgoing Data Filters* ("Validate Outgoing Data") – filters which inspect outgoing data to remote websites to ensure that it is appropriate to send and does not constitute part of an attack. Such data can be rewritten or blocked entirely depending on the filter.

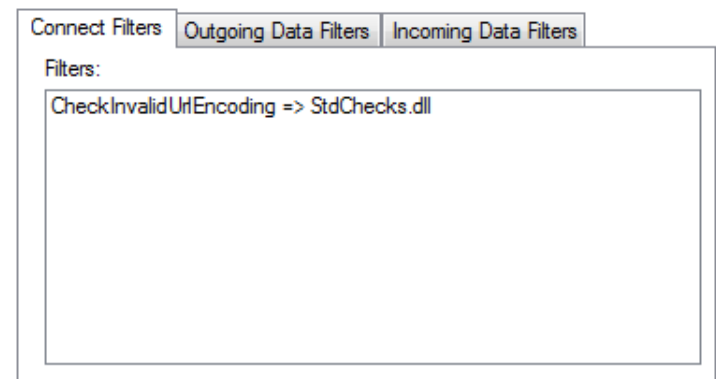
*Incoming Data Filters* ("Validate Incoming Data") – filters which inspect incoming data sent to the browser by websites. Such filters can rewrite the data before it reaches the browser or block it entirely if needs be.

It is possible to view filters which are currently enabled by navigating the three filters tabs:

Filters can be added or removed using the filter options under under the Manage Filters form, loaded by the Manage Filters button.



The screenshot shows a 'Manage Filters' form. At the top right is a 'Manage Filters' button. Below it is a section titled 'Feature Settings' with three checked checkboxes: 'Enable Feature', 'Validate Outgoing Connection', and 'Validate Incoming Data'. There is also a checkbox for 'Validate Outgoing Data'.



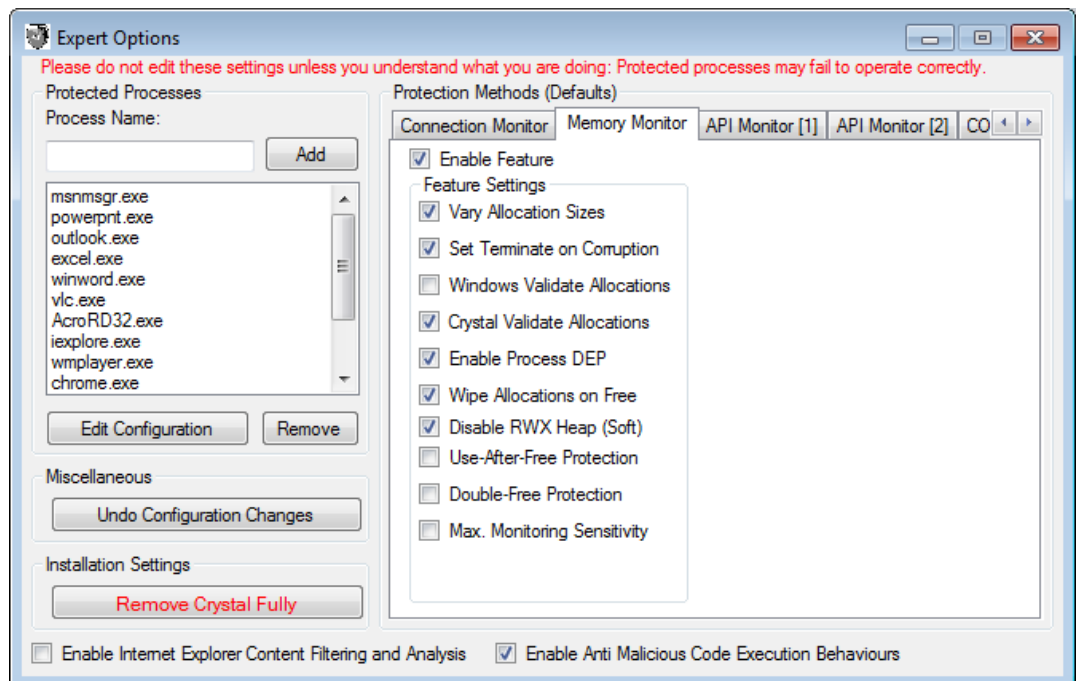
The screenshot shows the 'Connect Filters' tab of the filter management interface. It has three tabs: 'Connect Filters', 'Outgoing Data Filters', and 'Incoming Data Filters'. Below the tabs is a 'Filters:' label and a text area containing the filter rule: 'CheckInvalidUrlEncoding => StdChecks.dll'.

## Memory Monitor

The Memory Monitor category contains settings which allow for more fine-grained control over protection provided to protected program memory management routines. As a great number of security flaws are triggered by “memory corruption” flaws Crystal provides protection against this threat.

The individual settings presented in this category can be summarised as follows:

*Vary Allocation Sizes* – this setting allows protected programs to make their (heap) memory space far less predictable and hence harder to exploit by malicious software. This is generally a stable feature to enable as it does not attempt to detect threats, instead it simply increases entropy within the process memory space.



*Set Terminate on Corruption* – This feature enables a Windows feature which attempts to detect and terminate programs with corrupt (heap) memory. This feature is generally stable as it is implemented by the operating system and not by Crystal.

*Windows Validate Allocations* – This feature uses Windows functions to validate the layout of (heap) memory to help ensure that it is not corrupted, but is not – as far as the author is aware – designed with security in mind and hence may be more readily bypassed than the equivalent provided by Crystal, especially in older versions of the operating system. This feature should not generally be enabled as it results in a noticeable performance overhead.

*Crystal Validate Allocations* – This feature enables Crystal to validate that (heap) memory corruption has not occurred when a protected program attempts to request or return memory to the operating system. This should be quite effective in blocking certain types of memory corruption exploits. The performance overhead incurred by this protection method is not generally significant. The feature is fairly stable and should not result in problems for most applications.

*Wipe Allocations on Free* – This setting causes Crystal to automatically clear the contents of (heap) memory allocations when the memory is returned to the operating system by a protected program. This helps to mitigate against certain methods of exploiting “use-after-free” flaws, and adds a general layer of protection as sensitive information is cleared rather than retained in memory where it may otherwise be leaked to an attacker by an exploit.

*Disable RWX Heap (Soft)* – This setting enables Crystal to mark heap memory as not executable unless explicitly requested by the applications. The hope is that this will make it more difficult for software exploits to store malicious code in application memory and then to leverage a vulnerability to trigger execution of code from that memory. This isn't designed to mitigate return-oriented-programming attacks.

*Enable Process DEP* – This setting instructs Crystal to enable DEP (Data Execution Prevention) features for protected programs. DEP is an operating system provided service which includes further randomisation of the process memory space (at a more fundamental level than Crystal can stably achieve), and provides operating system support for blocking code running within suspect memory regions. This feature is generally stable, however older programs are occasionally not able to support DEP due to the way they are designed.

*Use-After-Free Protection* – This feature is provided by Crystal to mitigate cases in which memory which a program should have finished using is used after it is released back to the operating system. If memory is used in this way it is often possible to corrupt application behaviour due to the fact that the released memory is handed out to another aspect of the program which requires memory, and is then used by both separate aspects of the application (the original user and the new user), leading to a destabilising of the program which can be advantageous to attackers.

Such flaws are most generally observed being exploited in web browser software. It is recommended therefore that this feature be enabled for Internet Explorer when visiting high risk websites. This feature is stable, but may cause performance problems for memory intensive programs and is hence best enabled by default on computers with a large (more than 2GB) amount of RAM. *This feature can cause excessive memory consumption and may cause out-of-memory conditions for highly memory intensive applications.*

*Double-Free Protection* – This feature provides protection against a similar category of flaw as the Anti Use-After-Free issue, providing additional defences against this threat. The two protection settings should be enabled in tandem as there is no risk or stability issues associated with using both together. This setting will not do anything if Use-After-Free protection is not enabled.

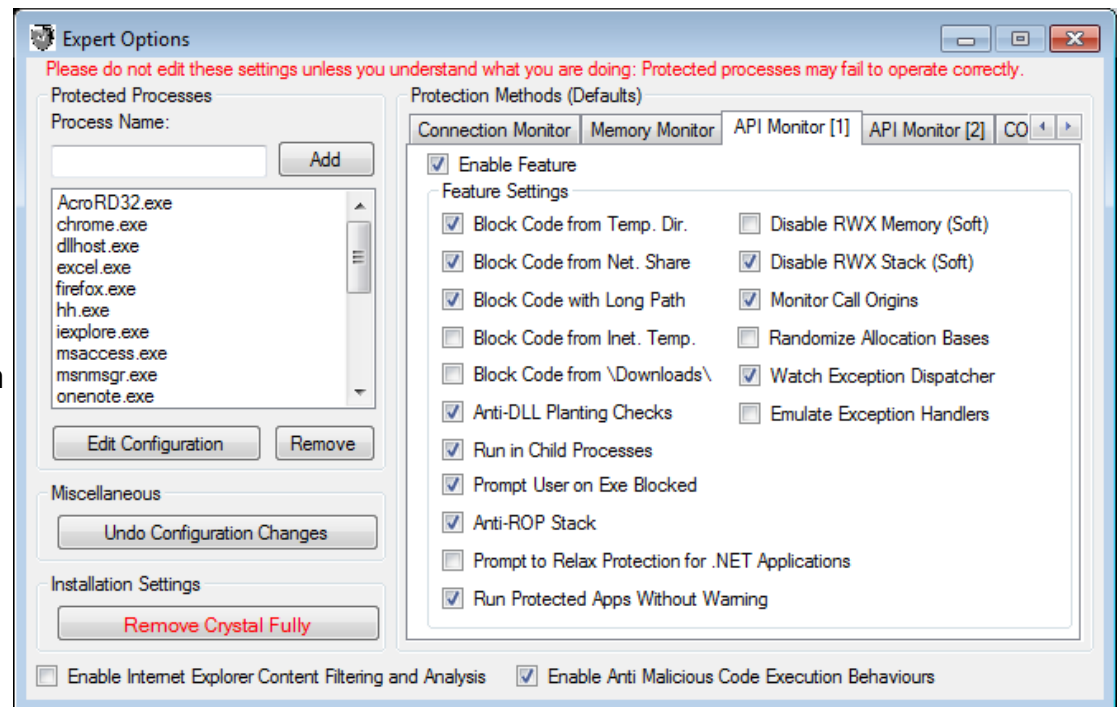
*Max. Monitoring Sensitivity* – This setting allows memory protection settings to be enabled in scenarios where they may otherwise crash or cause problems for a protected application. Generally this setting does not need to be enabled, and should not be enabled unless protected applications do not cope well with the settings listed above.

## API Monitor

The API Monitor category contains settings which apply to defences Crystal introduces to operating system functions to help detect and prevent abuse of these functions by malicious software. The category is also a mix of various protection methods which do not belong in other categories.

The individual settings presented in this category can be summarised as follows:

*Block Code from Temp. Dir.* – this setting instructs Crystal to block attempts by a protected program to load code modules or Programs from the temporary directory. The temporary directory can serve as a launch-pad for malicious code in certain attacks.



*Block Code from Net. Share* – This setting instructs Crystal to prevent code modules from being loaded from network shares (often a convenient method for an attacker to extend a local attack over the network).

*Block Code with Long Path* – This setting instructs Crystal to prevent loading of code modules from long paths. Many programs (antivirus, antimalware) have problems when attempting to process code residing in a folder with a long name, and hence this setting helps mitigate the risk posed.

*Block Code from Inet. Temp.* – This setting instructs Crystal to prevent loading of code modules from the Internet Explorer Temporary Files directory. This setting is disabled by default as it may impede the ability of legitimate downloaded software to run. In some cases however blocking this directory will reduce the risk of certain categories of exploit.

*Block Code from \Downloads\* - This setting instructs Crystal to prevent loading of code modules from folders with \Downloads\ in the name (a popular default for many applications). The idea is that occasionally legitimate software can be fooled into downloading and running software posing as updates or which exploit a flaw in DLL loading by update or install packages, and this setting helps to reduce the risk of these exploits succeeding.

*Anti-DLL Planting Checks* – A popular flaw exploited in many software packages caused by misguided attempts by the operating system to resolve the location of code modules required by the software when an associated file is opened (for example, .DOC files when Microsoft Word is opened). If the file in question

is hosted on a network share controlled by an attacker occasionally the operating system will include the network share as a valid location in which to search for the required code modules. If this occurs it provides a trivial avenue for attackers to run malicious code. This setting helps to protect against this type of threat.

*Run in Child Processes* – This setting enabled Crystal to load itself into all programs that are started by a protected application (32-bit programs only). This helps to protect against exploits which use a protected application as a launch vector for an attack targeting a less secure application (which the user may not realise is at risk – for example, Internet Explorer launching an obscure media player or other associated application).

*Prompt User on Exe Blocked* – This setting instructs Crystal to prompt a user whenever a program, launched by a protected application, is about to be blocked such as to give the user an opportunity to override Crystal and launch the program nonetheless. Crystal blocks all applications not on the executable whitelist by default.

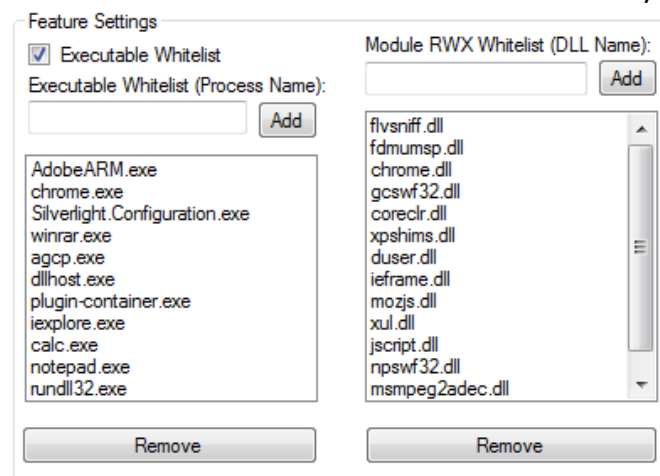
*Anti-ROP Stack* – This feature provides some protection against the exploit technique known as Return-Oriented Programming (ROP). As Windows provides (through DEP, to new Operating Systems such as Windows Vista and 7) methods for protecting against suspect memory locations from being executed as code, attackers occasionally have to craft malicious code using existing code fragments (for example fragments of trusted application code). To achieve this a complicated initial state much be achieved, often requiring memory which was not previously “stack memory” (i.e. execution state memory) to be used as



such. The anti-ROP stack helps to detect when this has happened and to terminate the program before a malicious payload can be properly run.

*Disable RWX Memory (Soft)* – This setting instructs Crystal that code running inside of the protected application is not able to request read/write-able and executable memory pages. This helps to protect against common vulnerability exploits which rely on the insertion of malicious code into writable memory before triggering its execution.

This feature is powerful in assisting with prevention of the execution of malicious code however it can interfere with application stability because certain software, written without security as a primary consideration, requires memory which is read/write-able and executable. To allow for compatibility with such software, while retaining the use of this feature of Crystal, the *Module RWX Whitelist* provides a customisable list of exceptions – DLLs which are able to request read/write-able and executable memory pages.



*Disable RWX Stack (Soft)* – This setting instructs Crystal to remove executable rights from process thread stack memory. This helps to protect against a popular software exploit technique in which malicious code is inserted into “stack” memory and is executed. This protection method should not have the same compatibility issues as the *Disable RWX Memory* method and is stable to use to protect most applications.

*Monitor Call Origins* – This feature enables Crystal to ensure that calls to certain key Windows operating system functions are being called from trusted memory locations. Malicious code executing from unrecognised memory locations – which invokes operating system functionality monitored by Crystal – will be terminated if this feature is enabled. This feature is generally stable and should not cause a problem for most applications.

*Randomize Allocation Bases* – This setting enables Crystal to perform additional randomisation of memory allocated within processes to help create an unpredictable environment for exploits attempting to run malicious code. Windows Vista and 7 provide similar a protection strategy by default to many applications, however some applications – as well as applications running on Windows XP – are not protected by the operating system. In these cases in particular this feature is a useful addition. This setting should be stable for most applications.

*Watch Exception Dispatcher* – This feature helps to protect against certain vulnerability exploitation techniques which rely on abusing the operating system exception handler structure residing in thread stack memory. Certain behaviours which are exhibited by the OS exception dispatcher are almost always indicative of a security flaw, and these are detected and mitigated by Crystal when this setting is enabled. Additional exception handler protection is provided by the operating system in Windows Vista and 7, however on some versions of XP this protection is absent or less robust, so this feature is a valuable addition to those OS.

*Emulate Exception Dispatcher* – This setting instructs Crystal to take further steps to determine whether an exception handler structure is being abused in a fashion that is typical for an exploit. This is achieved through basic emulation of the exception handler function. This feature acts as a backstop supporting the *Watch Exception Dispatcher* feature. In reality it is only in rare cases that this feature will add an additional layer of protection above that which is offered by the aforementioned feature, so enabling it is optional.

*Executable Whitelist* – The executable whitelist exists as a method for allowing certain programs to be started by protected programs without prompting or warning the user. This is mostly a user experience feature as it prevents unnecessary prompts from distracting the user, however if the *Prompt User on Exe Blocked* feature is not enabled then the Executable Whitelist constitutes the only means of a protected program being permitted to launch another program.

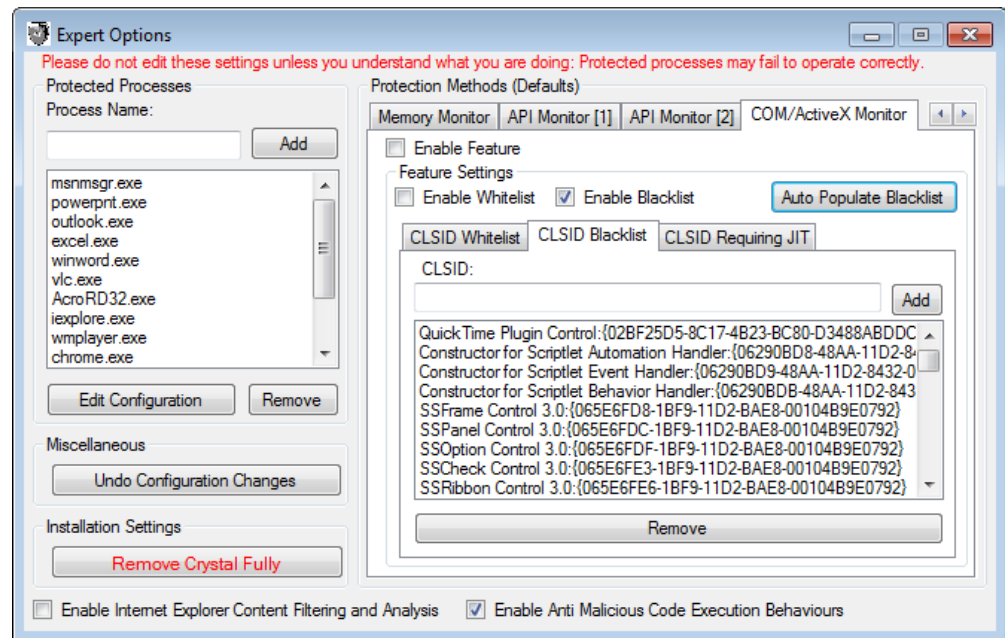
*Run Protected Apps Without Warning* – This setting enables Crystal to run applications on the protected processes list without prompting the user (for example, if Microsoft Word and Internet Explorer are both on the list, then Internet Explorer will be allowed to run Word without warning the user – of course this does not undermine the default behaviour, if a user would be warned by Internet Explorer normally they still will be).

## COM/ActiveX Monitor

The COM/ActiveX Monitor feature provides a method for Crystal to filter safe-for-scripting/initialisation ActiveX components on a per application basis.

Historically the exploitation of ActiveX vulnerabilities has been a major factor in contributing to Internet Explorer's poor security record.

This feature enables ActiveX control filtering on either a whitelist or blacklist basis. If the whitelist is enabled then only the listed ActiveX controls are permitted to instantiate within an application. This is the safest mode as it is far easier to determine which ActiveX controls are fairly safe than to exclude all those which are not. The blacklist mode of filtering blocks only select untrusted ActiveX controls from being instantiated.



The *Auto Populate Blacklist* button mines the Windows registry looking for ActiveX components which are safe-for-scripting/safe-for-initialisation and adds them all to the blacklist. Then a user is able to remove those controls which are trusted or needed and leave the remainder blocked, reducing the attack surface.

Entries in the whitelist/blacklist are listed in the following forms:

Control Name:{CLSID}  
{CLSID}

For example:

QuickTime Object:{02BF25D5-8C17-4B23-BC80-D3488ABDDC6B}

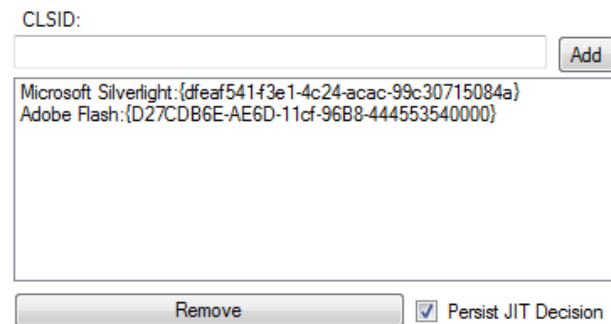
Or just:

{02BF25D5-8C17-4B23-BC80-D3488ABDDC6B}

The *CLSID Requiring JIT* provides a method for selective disabling of the API Monitor *Disable RWX Memory* setting whenever an attempt to load a class-ID on the respective list occurs. Certain ActiveX components such as Adobe Flash or Microsoft Silverlight require the ability to create executable memory to receive JIT (just-in-time) compiled code. As these controls are often implicated in exploits which bypass operating system provided defences due to their unique nature of requesting (attacker controlled) executable memory,

a user is prompted before any ActiveX control on this list is instantiated and is able to prevent the instantiation.

Being continually prompted when surfing the Internet can be a hindrance, so the *Persist JIT Decision* checkbox allows a user to persist their first decision (to allow or decline such ActiveX) throughout the remainder of their browser session (until the browser is closed).



The screenshot shows the 'Content Advisor' tab in the Internet Options dialog box. At the top, there is a 'CLSID:' label followed by an empty text box and an 'Add' button. Below this is a list box containing two entries: 'Microsoft Silverlight:{dfeaf541-f3e1-4c24-acac-99c30715084a}' and 'Adobe Flash:{D27CDB6E-AE6D-11cf-96B8-444553540000}'. At the bottom of the list box is a 'Remove' button. To the right of the list box is a checkbox labeled 'Persist JIT Decision', which is currently checked.

CLSID
Microsoft Silverlight:{dfeaf541-f3e1-4c24-acac-99c30715084a}
Adobe Flash:{D27CDB6E-AE6D-11cf-96B8-444553540000}

It should be noted that, for compatibility reasons, only ActiveX components which are marked safe-for-scripting/safe-for-initialisation (which means that web-based scripts can access their functionality directly – as is almost always required for the exploitation of an ActiveX-based vulnerability) are filtered. ActiveX which do not allow direct script control via the browser are not filtered.

# Troubleshooting

## **Why isn't Crystal compatible with certain software packages?**

As Crystal has only been tested for a limited amount of time with a limited number of software packages it is possible that there are certain applications which Crystal will not be able to work with effectively, generally due to either an overlap in behaviour (much as many antivirus packages will not run properly in tandem with other antivirus software installed on the same machine), or due to compatibility issues (an application requiring functionality Crystal classifies as potentially dangerous).

If such instances of incompatibility are discovered please do send a report to [admin@crystalaep.com](mailto:admin@crystalaep.com) containing any details of the application and the fault, and it will be investigated and if necessary changes will be made to help improve the product.

Certain antivirus programs are likely to detect the fact that Crystal is intercepting and altering the behaviour of operating system functionality to implement its protection strategies – a behaviour often also exhibited by malicious code seeking to hide itself from the user – and may incorrectly alert Crystal as a suspect application requiring an exemption to run.

In these cases please also contact the author and describe the fault and name the antivirus package which is blocking Crystal so that if necessary contact can be made with the AV company to have an exception added for Crystal or otherwise a workaround can be created.

### **How do I determine ideal protection level settings for a newly added program?**

Determining the ideal protection level for a program can be a bit of an art, as it is difficult to judge in advance whether a certain protection level may interfere with the normal operation of a protected program.

In general the High protection level provides the best balance between security enhancement for the protected application whilst avoiding the employment of protection methods which risk destabilising the protected application.

#### As a Basic User

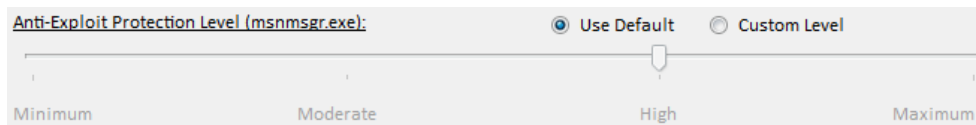
If an application is added for protection and does not respond well to the default protection level configured in the main user interface then it is suggested that the protection level be configured individually for that particular application.

Through the Basic Options form, set the protection level to Minimum, as follows:



1. Select the program short (executable) name in the Protected Programs list.

2. Notice the slider at the bottom of the Basic Options form:

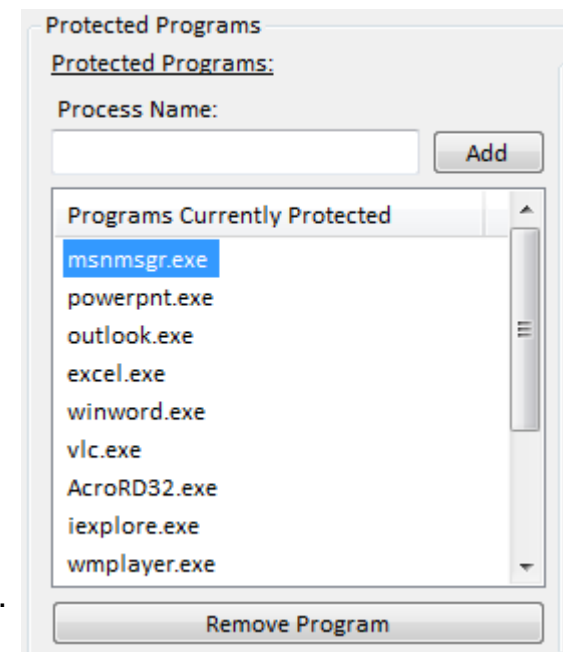


3. Select Custom Level, and drag the slider to minimum:



Ensure that you exit the program which is behaving abnormally (check that it does not show up in the Crystal main UI as a monitored process).

Re-start the program, it should be running with Minimum security settings. If it behaves abnormally still then remove it from the list of protected programs – Crystal is unable to protect the application. Otherwise if it behaves as expected, exit the program and then increase the security level to Moderate (in the fashion described earlier).



Re-start the program and determine whether it behaves abnormally. If it does not then repeat the process – moving the protection slider to High – and determine whether the program behaves properly at this setting. Otherwise reduce the protection level to the previous setting for that application and exit the Basic Options form. This setting is the application's ideal protection level.

### As an Expert User

To achieve the same on a more refined basis, set the application level to High and load the Expert Options form. Select the program you wish to protect and click *Edit Configuration*. Add/remove the protection methods offered under Memory Monitor and API Monitor, re-starting the program between modifications.

Please note that to save the changes made to a protected application in the Expert Options form you will need press the *Finished (Revert)* button. Eventually you will find the optimal configuration for the application at hand.

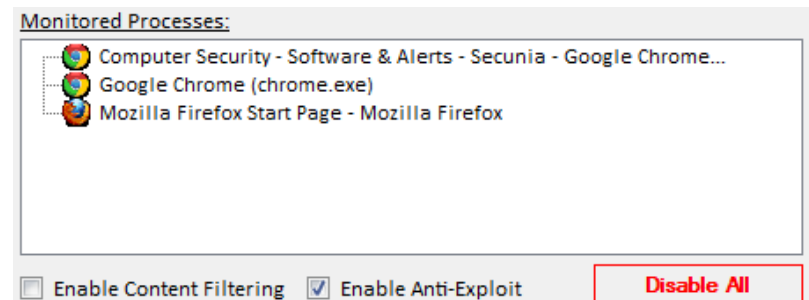
### Crystal prevents me from using certain webpages which I trust. How can I fix this?

Occasionally the content filters configured by Crystal may inaccurately recognise a webpage as unsafe. In these cases the easiest remedy is to close your browser (ensuring the browser is not displayed on the main interface Monitored Processes list) and to uncheck the *Enable Content Filtering* checkbox on the main interface form.

Re-start your browser and attempt the web page again, if Crystal was responsible for affecting its behaviour then it should work without problem the time round with content filtering disabled.

In the event that this does not fix the problem it may be advisable to try disabling Crystal entirely through the use of the Disable All button. Disabling all functionality in this way and then restarting Crystal will help you to conclusively determine whether Crystal is in any way responsible for the improper behaviour of the web page.

Remember to re-enable Crystal when you are done!



# Due Credit

As with any product many of the ideas and inspiration came directly from the IT security community in one form or another. The research projects, ideas and papers which contributed most directly to Crystal include:

- Matt Miller (skape) for his work on defending against the Structured Exception Handling overwrite exploitation method. The paper which contributed ideas is “Preventing the Exploitation of SEH Overwrites”[1].
- Piotr Bania for his work on defending against the Return-Oriented Programming exploitation method. The paper which contributed ideas is “Security Mitigations for Return-Oriented Programming Attacks”[2].
- Microsoft Security Research and Defense, and Matt Miller, for their work on the EMET software[3].
- Microsoft Research for their work on Microsoft Detours, ideas from which contributed towards the API interception functionality in Crystal (but not code, as the license is far expensive for the subset of functionality which Crystal requires).

[1] <http://uninformed.org/?v=5&a=2&t=txt>

[2] [http://piotrbania.com/all/articles/pbania\\_rop\\_mitigations2010.pdf](http://piotrbania.com/all/articles/pbania_rop_mitigations2010.pdf)

[3] <http://blogs.technet.com/b/srd/>

# About Crystal AEP

Crystal AEP is developed and maintained by Peter Winter-Smith ([admin@crystalaep.com](mailto:admin@crystalaep.com)). The product is written in C++ and C# .NET.

The product is at present closed source although this may be revised in the future depending on uptake and community support. The development API for implementing custom content filters is public and can be found at <http://www.crystalaep.com/development.html>

To contact the developer please use the email detailed above, and try to include "Crystal" in the subject line so that your email can be given priority.

The product website is <http://www.crystalaep.com>