

# **WPA Key Recovery**

# **Help Documentation**

Copyright © 2007 TamoSoft

## About WPA Key Recovery

WPA Key Recovery (WPAKR) is an add-on for CommView for WiFi designed to recover WPA passwords for WPA- or WPA2-protected 802.11 a/b/g wireless networks in Pre-Shared Key (PSK) mode. Because WPA/WPA2 encryption contains no known weaknesses, the recovery process is based on trying all passwords loaded from the dictionary file one by one, as well as optional password permutations.

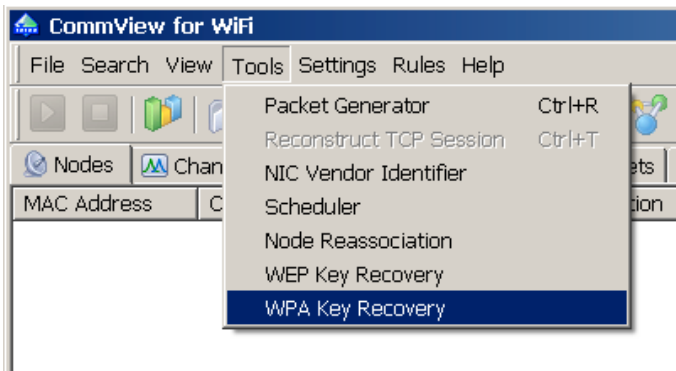
## Installation & Requirements

To install WPAKR, simply launch the setup file. WPAKR requires CommView for WiFi 5.6 or later and will be installed to the CommView for WiFi application folder. You can also install WPAKR on a computer without CommView for WiFi if you would like to use WPAKR in offline mode, for processing capture files copied from the computer running CommView for WiFi. If you install WPAKR on a computer without CommView for WiFi, you can launch it by double-clicking the program's executable file located in the folder to which you installed it.

If you'd like to use a distributed recovery process, i.e. use multiple computers simultaneously, you will find the installation instructions in the [Splitting the Job Between Multiple Computers](#) chapter.

## Usage

Once WPAKR has been installed, you can launch it by clicking **Tools => WPA Key Recovery** in the CommView for WiFi menu, as shown below:

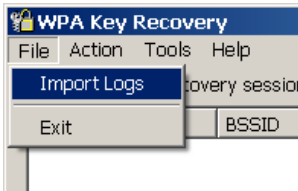


Note that this menu item becomes available only after you've installed WPAKR. When WPAKR is launched, CommView for WiFi establishes a TCP/IP connection with WPAKR so CommView for WiFi can send captured packets to WPAKR for analysis.

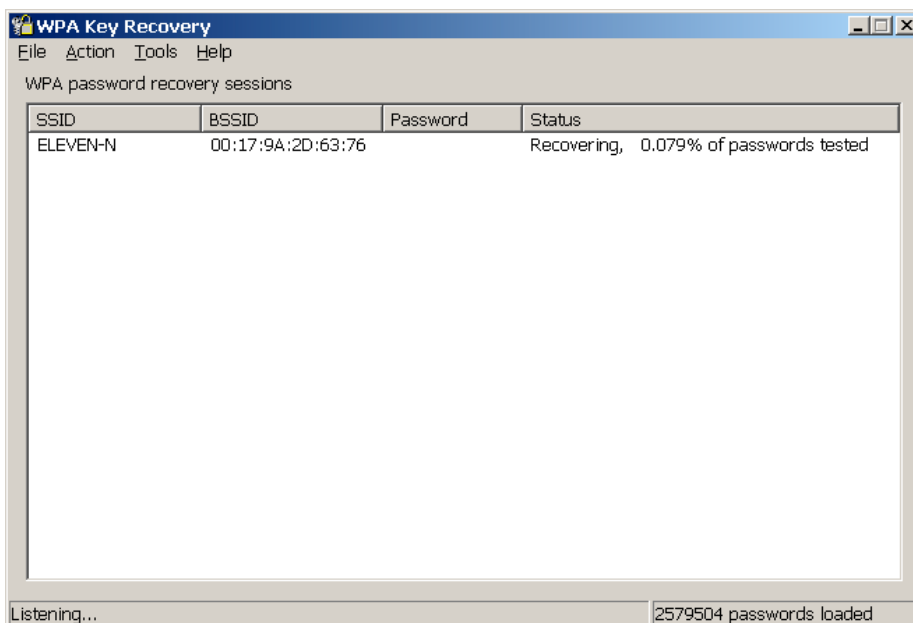
To be able to recover a WPA-PSK key, WPAKR **needs to receive packets with Association or Re-association Request followed by EAPOL key exchange packets**. These are the packets used in WPA for negotiating session keys. It's important that all of the EAPOL key exchange packets and at least one Association or Re-association Request packet be successfully captured. A damaged or missing EAPOL packet will make it impossible for WPAKR to start a key recovery process, and capturing the next EAPOL conversation between the AP and station may be required. This is an important distinction in the way WEP and WPA traffic is decrypted.

That said, WPAKR would display a new key recovery session only after CommView for WiFi has successfully captured a Association/Re-association Request packet followed by an EAPOL key exchange. This means that you should start capturing traffic from a WLAN in CommView for WiFi and wait for the next EAPOL exchange. EAPOL exchanges take place during the station association that may be triggered by connecting or reconnecting to the WLAN by the client, or restarting the AP, or by using the Node Reassociation tool in CommView for WiFi.

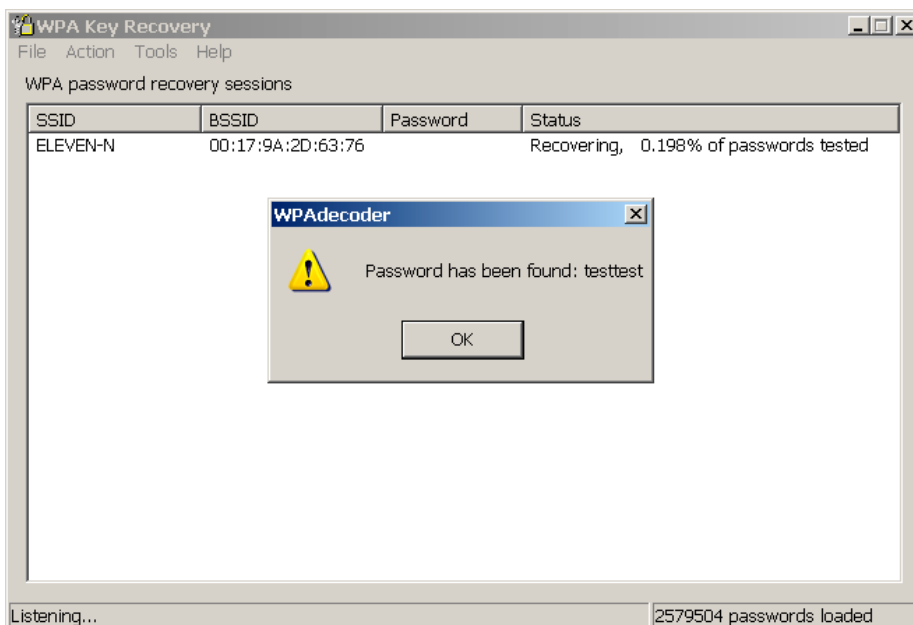
Alternatively, you can use WPAKR as a stand-alone application and import Association/Re-association Request and EAPOL packets previously captured by CommView for WiFi:



Once the necessary packets have been captured or loaded from a capture file, a new key recovery session will show up in WPAKR:



The **SSID** column lists the SSID of the access point. The **BSSID** column lists the hardware addresses of the access point. The **Password** column displays the recovered WPA password, if any. The **Status** column displays the current application status. Once the password has been recovered, a dialog box will display the password:



The obtained WPA key can be seen by clicking **Tools => WPA Passwords** in the application menu. WPAKR memorizes recovered keys between launches and tries them first, before attempting to recover them.

**Note that the evaluation version displays only the first two characters of the recovered password. The rest of the characters are replaced by asterisks. The licensed version displays all characters.**

The **Action** menu can be used for manually controlling the key recovery process.

### Recovery Speed, Dictionaries, and Password Permutations

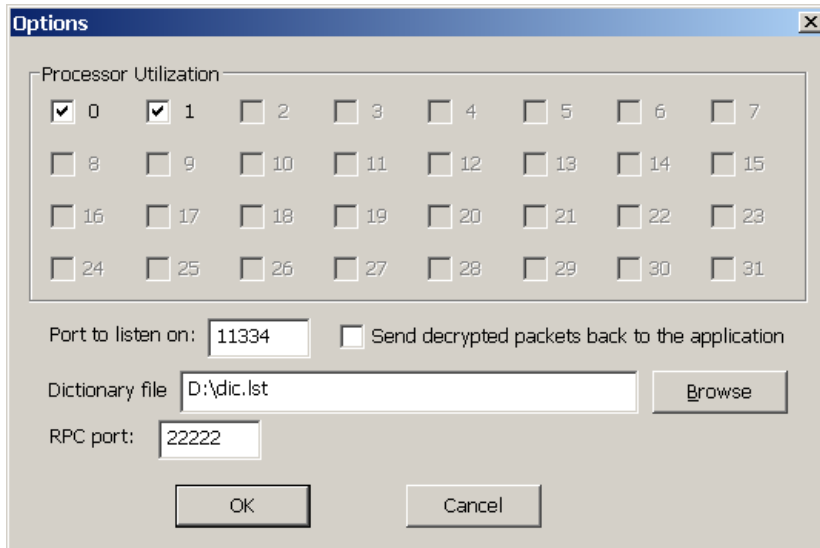
Because WPA uses robust encryption without known weaknesses, the only way to recover a password is by trying words from a dictionary file one by one. This process is very slow because each password must be hashed multiple times. A Pentium 4 2.8 GHz computer can try approximately 160 passwords per second. Because of such a low speed, a brute-force attack (i.e. trying all possible character combinations) doesn't make sense, as the minimum allowed WPA password length is eight characters. Trying all combinations even for a 5-character password will require  $90^5 = \sim 6$  billion attempts, or 1,000 days.

Given the 160 passwords per second benchmark, you can estimate the time it will take WPAKR to test all words from your dictionary file. For example, a dictionary file that contains one million passwords will be tested within two hours. It's important to understand that the WPA password will be recovered ONLY if this password can be found in your dictionary file. If a rare, hard-to-guess WPA password is selected; the chances that it will be found in the dictionary are slim. You can increase the recovery speed by [splitting the job between multiple computers](#).

To increase the chances of recovering the password, you should use a good and large dictionary file and, optionally, password permutations. Permutation is altering the passwords using the user-defined rules. For example, if the dictionary file contains the word "passWORD" the user may define mangling rules that will also test the word "password" and "PASSWORD." More information can be found in the [Password Permutations](#) chapter.

## Configuration

To configure WPAKR, click **Tools => Options**:



The screenshot shows the 'Options' dialog box for WPAKR. It features a 'Processor Utilization' section with 32 checkboxes, where checkboxes 0 and 1 are selected. Below this, there is a 'Port to listen on' field set to 11334, an unchecked checkbox for 'Send decrypted packets back to the application', a 'Dictionary file' field containing 'D:\dic.lst' with a 'Browse' button, and an 'RPC port' field set to 22222. The dialog concludes with 'OK' and 'Cancel' buttons.

**Processor Utilization** – for multi-core CPUs or multi-CPU computers, controls which processors or core(s) should be used for the computations. The key recovery process is very CPU-intensive, so you may want to limit the number of processors or processor cores to be used by the application. Restart the key recovery process to apply the changes.

**Port to listen on** – allows you to specify the TCP port that will be used for communicating locally with CommView for WiFi. If the default port (11334) is occupied by another application, you can change the port number.

**Send decrypted packets back to the application** – if this box is checked, WPAKR will "inject" the packets it has collected so far into CommView for WiFi once the key is recovered. Using this option is NOT recommended, as recovering the WPA key may take a long time during which all the captured packets will be buffered to the hard drive.

**Dictionary file** – the dictionary to be used for finding the WPA key. Any plain text file with one password per line will work. Remember that according to the WPA specifications, a WPA passphrase must be between 8 and 63 characters long and must contain only low ASCII characters (i.e. Latin characters, numbers, and special signs). The product comes with a short sample dictionary file. Big dictionaries are available to registered users. Restart the key recovery process to apply the changes.

**RPC port** – the port number to be used for a distributed recovery process that involves multiple computers. You will find more information about this setting in the [Splitting the Job Between Multiple Computers](#) chapter.

## Splitting the Job Between Multiple Computers

WPAKR can use multiple computers simultaneously for distributed password recovery. In this mode, one computer works as a server that distributes the tasks between other computers (clients), while participating in password breaking itself. The mode of operation requires Windows XP SP2 or higher.

To configure job splitting, do the following:

- On the server machine, click **Start** => **Run** and execute gpedit.msc. A policy configuration window will appear. Select **Computer Configuration** => **Administrative Templates** => **System** => **Remote Procedure Call** => **Restrictions for Unauthenticated RPC Clients**. Select "Enabled" and then "None" in the drop-down list. Reboot the computer. This will enable communication between the server and clients using Remote Procedure Calls (RPC).
- Launch WPAKR and configure the RPC port in **Tools** => **Options** (by default, 22222). If you use a firewall, make sure that incoming TCP connections to this port are permitted. Assign a job to WPAKR (either by importing an NCF file or by running CommView for WiFi and letting WPAKR get the EAPOL key exchange packets from CommView for WiFi).
- Find WPAclient.exe (that's the client part of the application) in the folder to which WPAKR was installed and copy it to the client computers (up to 64 clients are allowed). Additionally, you must copy exactly the same dictionary file that you use in WPAKR to the same folder to which WPAclient.exe was copied. The dictionary file name must be "dictionary.lst."
- Launch WPAclient.exe, then click **Tools** => **Options** and enter the IP address or hostname of the server and the RPC port. The client(s) will then connect to the server and receive the jobs from it.

## Password Permutations

In addition to simple testing of passwords from the dictionary files, WPAKR is capable of altering the passwords using the user-defined rules. For example, if the dictionary file contains the word "passWORD", the user may define mangling rules that will also test the word "password" and "PASSWORD."

To control password permutations, edit the word mangling rules by clicking **Tools => Mangling rules** in WPAKR. WPAKR comes with a set of pre-defined rules. Inactive rules are commented out using the "#" sign. By default, only the "try words as they are" rule is enabled; all other ones are commented out. You can uncomment other rules or compose your own rules. The rules syntax is described in detail below.

Note that using rules increases your chances for successful key recovery but, at the same time, slows down the recovery speed. For example, a rule that tells the program to prefix all words with a single-digit number will make the recovery process ten times slower.

The password permutations engine is based on the code licensed from John the Ripper project, Copyright © Solar Designer.

### Rule Syntax

Each wordlist rule consists of optional rule reject flags followed by one or more simple commands, listed all on one line and optionally separated with spaces. There's also a preprocessor, which generates multiple rules for a single source line. Below you will find descriptions of the rule reject flags, the rule commands, and the preprocessor syntax.

#### Character position codes

Character positions are numbered starting with 0, and specified in rules with the following characters:

0...9	for 0...9
A...Z	for 10...35
*	for max_length
-	for (max_length - 1)
+	for (max_length + 1)

Here, max\_length is the maximum plaintext length supported for the current hash type.

The same characters are also used for specifying other numeric parameters.

#### Character classes

??	matches "?"
?v	matches vowels: "aeiouAEIOU"
?c	matches consonants: "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTUVWXYZ"
?w	matches whitespace: space and horizontal tabulation characters
?p	matches punctuation: ".,:;!'" and the double quote character
?s	matches symbols "\$%^&*()-_+= \<>[]{}#~/~"
?l	matches lowercase letters [a-z]
?u	matches uppercase letters [A-Z]
?d	matches digits [0-9]
?a	matches letters [a-zA-Z]
?x	matches letters and digits [a-zA-Z0-9]

The complement of a class can be specified by uppercasing its name. For example, "?D" matches everything but digits.

#### Simple commands

:	no-op: do nothing to the input word
l	convert to lowercase
u	convert to uppercase
c	capitalize
C	lowercase the first character, and uppercase the rest
t	toggle case of all characters in the word
TN	toggle case of the character in position N
r	reverse: "Fred" -> "derF"
d	duplicate: "Fred" -> "FredFred"
f	reflect: "Fred" -> "FredderF"

```
{ rotate the word left: "jsmith" -> "smithj"
} rotate the word right: "smithj" -> "jsmith"
$X append character X to the word
^X prefix the word with character X
```

#### Length control commands

```
<N reject the word unless it is less than N characters long
>N reject the word unless it is greater than N characters long
'N truncate the word at length N
```

#### English grammar commands

```
p pluralize: "crack" -> "cracks", etc. (lowercase only)
P "crack" -> "cracked", etc. (lowercase only)
l "crack" -> "cracking", etc. (lowercase only)
```

#### Insert/delete commands

```
[ delete the first character
] delete the last character
DN delete the character in position N
xNM extract substring from position N for up to M characters
iNX insert character X in position N and shift the rest right
oNX overstrike character in position N with character X
```

Note that square brackets ("[" and "]") are special characters to the preprocessor: you should escape them with a backslash ("\") if using these commands.

#### Charset conversion commands

```
S shift case: "Crack96" -> "cRACK(^"
V lowercase vowels, uppercase consonants: "Crack96" -> "CRaCK96"
R shift each character right, by keyboard: "Crack96" -> "Vtsvl07"
L shift each character left, by keyboard: "Crack96" -> "Xeaxj85"
```

#### Memory access commands

```
M memorize the word
Q reject the word unless it has changed
```

#### Character class commands

```
sXY replace all characters X in the word with Y
s?CY replace all characters of class C in the word with Y
@X purge all characters X from the word
@?C purge all characters of class C from the word
!X reject the word if it contains character X
!?C reject the word if it contains a character in class C
/X reject the word unless it contains character X
/?C reject the word unless it contains a character in class C
=NX reject the word unless character in position N is equal to X
=N?C reject the word unless character in position N is in class C
(X reject the word unless its first character is X
(?C reject the word unless its first character is in class C
)X reject the word unless its last character is X
)?C reject the word unless its last character is in class C
%NX reject the word unless it contains at least N instances of X
%N?C reject the word unless it contains at least N characters of class C
```

#### Extra "single crack" mode commands

When defining "single crack" mode rules, extra commands are available for word pairs support, to control if other commands are applied to the first, the second, or to both words:

- 1 first word only
- 2 second word only
- + the concatenation of both (should only be used after a "1" or "2")

If you use some of the above commands in a rule, it will only process word pairs (e.g., full names from the GECOS field) and reject single words. A "+" is assumed at the end of any rule that uses some of these commands, unless you specify it manually. For example, "112u" will convert the first word to lowercase, the second one to uppercase, and use the concatenation of both. The use for a "+" might be to apply some more commands: "112u+r" will reverse the concatenation of both words, after applying some commands to them separately.

#### The rule preprocessor

The preprocessor is used to combine similar rules into one source line. For example, if you need to make WPAKR try lowercased words with digits appended, you could write a rule for each digit, 10 rules total. Now imagine appending two-digit numbers -- the configuration file would get large and ugly.

With the preprocessor you can do these things easier. Simply write one source line containing the common part of these rules followed by the list of characters you would have put into separate rules, in square brackets (the way you would do in a regexp). The preprocessor will then generate the rules for you. For the examples above, the source lines will be "[l\$[0-9]" (lowercase and append a digit) and "[l\$[0-9]\$[0-9]" (lowercase and append two digits). These source lines will be expanded to 10 and 100 rules, respectively. By the way, preprocessor commands are processed right-to-left while character lists are processed left-to-right, which results in normal ordering of numbers in the above examples and in other typical cases. Note that arbitrary combinations of character ranges and character lists are valid. For example, "[aeiou]" will use vowels and "[aeiou0-9]" will use vowels and digits. If you need to have WPAKR try vowels followed by all other letters, you can use "[aeioua-z]" -- the preprocessor is smart enough to not produce duplicate rules in such cases.

There are some special characters in rules ("[" starts a preprocessor character list, "-" marks a range inside the list, etc.). You should prefix them with a backslash ("\") if you want to put them inside a rule without using their special meaning. Of course, the same applies to "\" itself. Also, if you need to start a preprocessor character list at the very beginning of a line, you'll have to prefix it with a ":" (the no-op rule command), or it would be treated as a new section start.

## Frequently Asked Questions

**Q. Can WPAKR recover WPA keys in non-PSK modes?**

A. No.

**Q. Do you use pre-computed hash tables a.k.a. rainbow tables?**

A. No.

**Q. So how long does it take to recover a WPA-PSK key?**

A. From one second to never. It all depends on the key and the dictionary file that you use. If the key being sought can be found in your dictionary file (you can also use permutations), then the time needed to recover it will be proportional to the position of the key in the dictionary. For example, if the key is on the one-millionth position in the dictionary, it will be found in approximately two hours. But if your dictionary doesn't contain the password in question (even with permutations), then the WPA key won't be found. Recovering a WPA key is different from recovering a WEP key, where no dictionary is needed and success is guaranteed if you have collected a few hundred thousand packets.

## How to Purchase WPA Key Recovery

This program is a 30-day evaluation version.

A fully functional, unrestricted version of the program can be purchased for US\$999. Call us for pricing on larger site licenses. As a registered customer, you are entitled to:

- Free updates that will be released within 1 year from the date of purchase
- Information on updates and new products
- Free technical support

We accept credit card orders, orders by phone and fax, checks, and wire transfers. Prices, terms, and conditions are subject to change without notice.

## Contacting Us

Questions? Comments? Suggestions? Bug reports? Don't hesitate to contact us.

<http://www.tamos.com/>

When describing your problem, please be as specific as possible. A detailed description of the problem will help us solve it much faster. Please don't forget to mention your OS version, the program version and build (**Help** => **About**), and any other details that you think may be relevant.

## Other Products

### CommView

CommView is a program for monitoring Internet and Local Area Network (LAN) activity and is capable of capturing and analyzing network packets. It gathers information about data passing through your dial-up connection or Ethernet card and decodes the analyzed data. With CommView you can see the list of network connections and vital IP statistics and examine individual packets. Packets are decoded down to the lowest layer with full analysis of the most widespread protocols. Full access to raw data is also provided in real time. CommView is a helpful tool for LAN administrators, security professionals, network programmers, or anyone who wants to have a full picture of the traffic going through one's PC or LAN segment.

[More information](#)

### CommTraffic

CommTraffic is a network utility for collecting, processing, and displaying traffic and network utilization statistics for network connections, including LAN and dial-up. It shows traffic and network utilization statistics for each computer in the segment. The software provides a very attractive and customizable interface, with an optional tray icon menu that displays general network statistics. You can also generate reports that reflect the network traffic volume and Internet connection expenses (if any). CommTraffic supports virtually any rate plan your ISP might use, such as the one based on connection time, traffic volume, time of the day, and other measures. You can set alarms that will inform you when certain criteria (e.g. amount of traffic, expenses) are reached. A configuration wizard will guide you through the setup and automatically detect your network or connection settings.

[More information](#)

### SmartWhois

SmartWhois is a handy utility for obtaining information about any IP address, hostname, or domain in the world. Unlike standard whois utilities, it automatically delivers information associated with an IP address or domain no matter where it is registered geographically. In just a few seconds, you get all you want to know about a user: domain, network name, country, state or province, and city. Even if the IP address cannot be resolved to a hostname, SmartWhois won't fail!

[More information](#)

### CountryWhois

CountryWhois is a utility for identifying the geographic location of an IP address. CountryWhois can be used to analyze server logs, check e-mail address headers, identify online credit card fraud, or in any other instance where you need to quickly and accurately determine the country of origin by IP address.

[More information](#)

### Essential NetTools

Essential NetTools is a set of network tools useful in diagnosing networks and monitoring your computer's network connections. It's a Swiss Army knife for everyone interested in a set of powerful network tools for everyday use. The program includes a NetStat utility that shows your computer's network connections and open ports and maps them to the owning application. It also features a fast NetBIOS scanner, a NetBIOS Auditing Tool for checking LAN security, and a monitor of external connections to your computer's shared resources, as well as a process monitor that displays information about all the programs and services running on your computer. Other useful tools are included, such as Ping, TraceRoute, and NSLookup. Additional features include report generation in HTML, text, and comma delimited formats and a customizable interface. The program is an easy-to-use and powerful replacement for such Windows utilities as nbtstat, netstat, and NetWatcher. It incorporates many advanced features that standard Windows tools can't offer.

[More information](#)

### DigiSecret

DigiSecret is an easy-to-use, secure, and powerful application for file encryption and sharing. It utilizes strong and time-proven encryption algorithms for creating encrypted archives, self-extracting EXE files, and sharing files with your associates and friends. DigiSecret also includes powerful and intelligent file compression; you no longer need .zip files when you can have encrypted and compressed DigiSecret files. The program is integrated with the Windows shell, and you can perform operations on files by right-clicking on them. It also fully supports drag-and-drop operations.

[More information](#)

### **NetResident**

NetResident is an advanced network content monitoring program that captures, stores, analyzes, and reconstructs network events such as e-mail messages, Web pages, downloaded files and instant messages. NetResident uses advanced monitoring technology to capture the required data from the network, saves it to a database, reconstructs it, and displays this content in an easy-to-understand format. While NetResident is similar to network analyzers in many respects, it focuses on high-level protocols that are used to transfer content over the Internet or LAN. With NetResident, you don't need a profound understanding of networking technologies, have to use complex packet capturing and analysis software, or dig through network packets in order to reconstruct the actual data; NetResident does all the work for you and presents only the Web pages, e-mails, instant messages, or downloaded files as requested. NetResident is used by network administrators to enforce IT policy, parents to monitor their children's communication on the Internet, and by forensic experts to gain crucial information.

[More information](#)