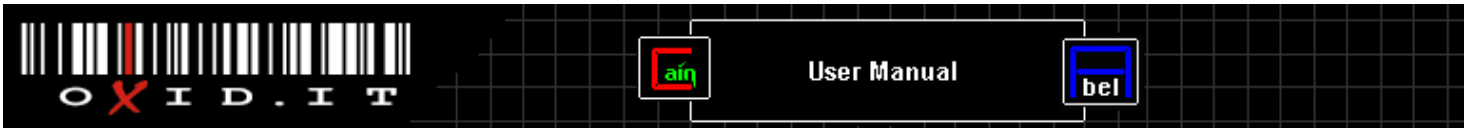


Introduction

Cain & Abel - User Manual

web site: http://www.oxid.it
email: mao@oxid.it

Cain & Abel is a password recovery tool for Microsoft Operating Systems. It allows easy recovery of several kind of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols.

The program does not exploit any software vulnerabilities or bugs that could not be fixed with little effort. It covers some security aspects/weakness intrinsic of protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "non standard" utilities for Microsoft Windows users.

Cain & Abel has been developed in the hope that it will be useful for network administrators, teachers, security consultants/professionals, forensic staff, security software vendors, professional penetration testers and everyone else that plans to use it for ethical reasons. The author will not help or support any illegal activity done with this program.

Be warned that there is the possibility that you will cause damages and/or loss of data using this software and that in no events shall the author be liable for such damages or loss of data. Please carefully read the License Agreement included in this manual before using the program.

License Agreement

You should carefully read the following terms and conditions before using this software. Your use of this software indicates your acceptance of this license agreement. If you do not agree to all of the terms of this agreement, do not use this software.

Title and Ownership

This software is owned by Massimiliano Montoro (mao@oxid.it), you may not reverse engineer, disassemble or de-compile this software nor may you permit anyone else to do so.

FREEWARE Version

You are hereby licensed to: use the FREEWARE Version of the software forever; make as many copies of the FREEWARE version of this software and documentation as you wish; give exact copies of the original FREEWARE version to anyone; and distribute the FREEWARE version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above. You are specifically prohibited from charging or requesting donations for any of such copies and from distributing the software/documentation with other products (commercial or otherwise) without prior written permission.

Disclaimer Of Warranty and Limitation of Liability

THIS SOFTWARE AND THE ACCOMPANYING FILES (IF ANY) ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHOR DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ANY DEFECTS DISCOVERED IN THE SOFTWARE WILL BE CORRECTED. FURTHERMORE, THE AUTHOR DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. YOU ASSUME ALL RISKS IN USING THE SOFTWARE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY THE AUTHOR OR ANY OF THEIR AUTHORIZED REPRESENTATIVES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS OR SPECIAL DAMAGES, EVEN IF THE AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. CERTAIN STATES DO NOT PERMIT EXCLUSIONS OF IMPLIED WARRANTIES OR LIMITATIONS OF LIABILITY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU OR MAY APPLY TO YOU ONLY IN PART. YOU MAY HAVE OTHER LEGAL RIGHTS WHICH VARY FROM STATE TO STATE.

Installation

Cain & Abel is a two part program distributed at http://www.oxid.it as a Self-Installing executable package named "ca_setup.exe".
Cain (Cain.exe) is the main GUI application, Abel is a Windows service composed by two files: Abel.exe and Abel.dll.

Requirements

The actual version requires the following items:
- 10Mb Hard-Disk space
- Microsoft Windows 2000/XP/2003/Vista
- Winpcap Packet Driver (v2.3 or above; AirPcap adapter is supported from Winpcap version 4.0). Please check the documentation on their site.
- Airpcap Packet Driver (for passive wireless sniffer / WEP cracker).

Setup

Just run the Self-Installing executable package and follow the installation instructions.
The package will copy all the files needed by the program into the installation directory.

Installation Files

Cain's setup program will install and/or replace these files in your system:

- Cain.exe [the main executable program]
- Cain.exe.sig [author's PGP signature of the file Cain.exe]
- CA_UserManual.chm [this file]
- Abel.exe [the executable of the Windows service named Abel]
- Abel.exe.sig [author's PGP signature of the file Abel.exe]
- Abel.dll [a DLL file needed by the program]
- Abel.dll.sig [author's PGP signature of the file Abel.dll]
- Uninstal.exe [the uninstallation program]
- Wordlist.txt [a little word list file]
- Install.log [the log file of the installation package, you can check everything modified on your system here]
- Whatsnew.txt [contains differences between versions]
- oui.txt [a list file that contains vendor's information about MAC addresses]
- <Installation Dir>\winrtgen\winrtgen.exe [Winrtgen - a windows utility to generate Rainbow Tables]
- <Installation Dir>\winrtgen\winrtgen.exe.sig [author's PGP signature of the file winrtgen.exe]
- <Installation Dir>\winrtgen\charset.txt [an example file containing charset definitions for winrtgen.exe and Cain's cryptanalysis attacks]
- <Installation Dir>\Driver\WinPcap_4_1_beta5.exe [the original distribution package of the Winpcap drivers]

All the above files will be installed in the Installation directory and subdirectories.

Abel Installation

Abel is a Windows NT service composed of two files: "Abel.exe" and "Abel.dll". These files are copied by the installation package into the program's directory but the service is NOT automatically installed on the system.
Abel can be installed locally or remotely (using Cain) and requires Administrator's privileges on the target machine.

LOCAL INSTALLATION:
1) Copy the files Abel.exe and Abel.dll into the %WINNT% directory (E.G.: C:\WINNT or C:\Windows)
2) Launch Abel.exe to install the service (it is not automatically started)
3) Start the service using the Windows Service Manager (services.msc)

REMOTE INSTALLATION:
1) Use the "Network TAB" in Cain and choose the target remote computer where you want to install Abel
2) Right click on the computer icon in the left tree and select "Connect As"
3) Provide Administrator's credentials for the remote system
4) Once connected right click on the "Services" icon and select the menu entry "Install Abel"
5) That's it! The two files ‘Abel.exe’ and ‘Abel.dll’ will automatically be copied to the remote machine’s root directory i.e. C:\winnt, C:\Windows); the service will automatically be installed and started.

Registry Modifications

Like any other setup program Cain involves making changes to your registry. Whenever registry changes are made, it is always advisable to backup your registry first.
Cain's settings are all located under the HKEY_CURRENT_USER\Software\Cain registry key.

Dependencies

Cain.exe depends on or requires the following libraries: Abel.dll, Crypt32.dll, Pstorec.dll, Kernel32.dll, Advapi32.dll, Comctl32.dll, Comdlg32.dll, Gdi32.dll, Iphlpapi.dll, Mpr.dll, NetApi32.dll, Odbc32.dll, Ole32.dll, Oleaut32.dll, Packet.dll (Winpcap), Rasapi32.dll, Rpcrt4.dll, Shell32.dll, User32.dll, Wpcap.dll (Winpcap), Airpcap.dll (AirPcap), Ws2_32.dll, Wsnmp32.dll.

Abel.exe depends on or requires the following libraries: Abel.dll, Kernel32.dll, Advapi32.dll, Iphlpapi.dll, User32.dll, Ws2_32.dll.

Abel.dll depends on or requires the following libraries: Lsasrv.dll, Kernel32.dll, Advapi32.dll, User32.dll, samsrv.dll.

Post installation generated files

Cain will create the following files (comma separated list files) in the program's installation directory:

Cracker
- APOP-MD5.LST [contains a list of credentials of type APOP-MD5]
- CRAM-MD5.LST [contains a list of credentials of type CRAM-MD5]
- PIX-MD5.LST [contains a list of credentials of type Cisco PIX]
- IOS-MD5.LST [contains a list of credentials of type Cisco IOS]
- PWLS.LST [contains a list of PWL files and relative credentials]
- NTLMv2.LST [contains a list of credentials of type NTLMv2]
- LMNT.LST [contains a list of credentials of type LM & NTLMv1]
- CACHE.LST [contains a list of credentials of type MS-CACHE]
- OSPF-MD5.LST [contains a list of credentials of type OSPF-MD5]
- RIP-MD5.LST [contains a list of credentials of type RIPv2-MD5]
- VRRP-HMAC.LST [contains a list of credentials of type VRRP-HMAC]
- VNC-3DES.LST [contains a list of credentials of type VNC Triple DES]
- MD2.LST [contains a list of hashes of type MD2]
- MD4.LST [contains a list of hashes of type MD4]
- MD5.LST [contains a list of hashes of type MD5]
- SHA-1.LST [contains a list of hashes of type SHA-1]
- SHA-2.LST [contains a list of hashes of type SHA-2]
- RIPEMD-160.LST [contains a list of hashes of type RIPEMD-160]
- K5.LST [contains a list of credentials of type Ms-Kerberos PreAuth]
- RADIUS_SHARED_HASHES.LST [contains a list of credentials of type RADIUS PreShared Key]
- IKEPSKHashes.LST [contains a list of credentials of type IKE-PSK]
- MSSQLHashes.LST [contains a list of credentials of type Microsoft SQL]
- MySQL.LST [contains a list of credentials of type MySQL]
- ORACLE.LST [contains a list of credentials of type ORACLE]
- TNS-HASHES.LST [contains a list of credentials of type ORACLE-TNS]
- 80211.LST [contains a list of 802.11 capture files]
- SIPHASHES.LST [contains a list of hashes used in SIP protocol]
- TOKENS.LST [contains a list of RSA token serial numbers and seeds]
- WPAPSK.LST [contains a list of hashes of type WPA-PSK]
- CHAP.LST [contains a list of hashes of type CHAP-MD5]

Sniffer
- HOSTS.LST [contains a list of host's information such as MAC address, IP address, Hostnames]
- APR.LST [contains a list of hosts to be used in APR]
- DRR.LST [contains a list of host names and IP addresses to be used by APR-DNS]
- SSH-1.LST [contains references to files generated by SSH-1 sniffer filter]
- CERT.LST [contains references to certificate files to be used by APR-HTTPS]
- HTTPS.LST [contains references to files generated by APR-HTTPS sniffer filter]
- FTPS.LST [contains references to files generated by APR-FTPS sniffer filter]
- IMAPS.LST [contains references to files generated by APR-IMAPS sniffer filter]
- LDAPS.LST [contains references to files generated by APR-LDAPS sniffer filter]
- POP3S.LST [contains references to files generated by APR-POP3S sniffer filter]
- RDP.LST [contains references to files generated by APR-RDP sniffer filter]
- FTP.LST [contains a list of credentials captured by FTP sniffer filter]
- HTTP.LST [contains a list of credentials captured by HTTP sniffer filter]
- IMAP.LST [contains a list of credentials captured by IMAP sniffer filter]
- POP3.LST [contains a list of credentials captured by POP3 sniffer filter]
- SMB.LST [contains a list of credentials captured by Server Message Block sniffer filter]
- TELNET.LST [contains references to files generated by Telnet sniffer filter]

- VNC.LST [contains a list of credentials captured by VNC sniffer filter]
- TDS.LST [contains a list of credentials captured by TDS (Tabular Data Stream) sniffer filter]
- SMTP.LST [contains a list of credentials captured by SMTP sniffer filter]
- NNTP.LST [contains a list of credentials captured by NNTP sniffer filter]
- KRB5.LST [contains a list of credentials captured by MS-Kerberos5 sniffer filter]
- DCERPC.LST [contains a list of credentials captured by DCE/RPC sniffer filter]
- RADIUS.LST [contains a list of pre shared keys captured by RADIUS sniffer filter]
- RADIUS_USERS.LST [contains a list of user's credentials captured by RADIUS sniffer filter]
- ICQ.LST [contains a list of credentials captured by ICQ sniffer filter]
- IKE-PSK.LST [contains a list of pre shared keys captured by IKE sniffer filter]
- MySQL.LST [contains a list of credentials captured by MySQL sniffer filter]
- SNMP.LST [contains a list of community strings captured by SNMP sniffer filter]
- VoIP.LST [contains a list of VoIP conversations captured by SIP/RTP sniffer filter]
- WPAPSKAUTH.LST [contains a list of credentials captured by WPAPSK sniffer filter]
- TNS.LST [contains a list of user's credentials captured by ORACLE-TNS sniffer filter]
- GRE_PPP.LST [contains a list of user's credentials captured by GRE/PPP sniffer filter]
- PPPoE.LST [contains a list of user's credentials captured by PPPoE sniffer filter]

Other files
- RT.LST [contains the list of Rainbow Tables to use during Cryptanalysis attacks]
- QLIST.LST [contains hosts of the quick list in the Network Tab]
- CCDU.LST [contains information about Cisco Config Downloader/Uploader View]
- HTTP_USER_FIELDS.LST [contains a list of user name fields to be used by the HTTP-FORM and HTTP-COOKIE sniffer filter]
- HTTP_PASS_FIELDS.LST [contains a list of password fields to be used by the HTTP-FORM and HTTP-COOKIE sniffer filter]
- DUMP.IVS [contains a list of WEP IVs in aircrack-ng's compatible format]

Subdirectories
- <Installation Dir>\Certs\ [contains fake certificate files (*.crt) to be used by APR SSL spoofing sniffers]
- <Installation Dir>\HTTPS\ [contains session files captured by APR-HTTPS]
- <Installation Dir>\FTPS\ [contains session files captured by APR-FTPS]
- <Installation Dir>\POP3S\ [contains session files captured by APR-POP3S]
- <Installation Dir>\IMAPS\ [contains session files captured by APR-IMAPS]
- <Installation Dir>\LDAPS\ [contains session files captured by APR-LDAPS]
- <Installation Dir>\SSH-1\ [contains session files captured by APR-SSH-1]
- <Installation Dir>\Telnet\ [contains session files captured by Telnet sniffer filter]
- <Installation Dir>\VoIP\ [contains VoIP conversations captured by the sniffer and saved as WAV files]
- <Installation Dir>\CCDU\ [contains configurations files from Cisco devices]

Uninstallation

You can uninstall Cain from the Add/Remove Programs tool in the Control Panel or by directly executing the unistallation program. The uninstall program will not remove the Abel service.

Abel Uninstallation

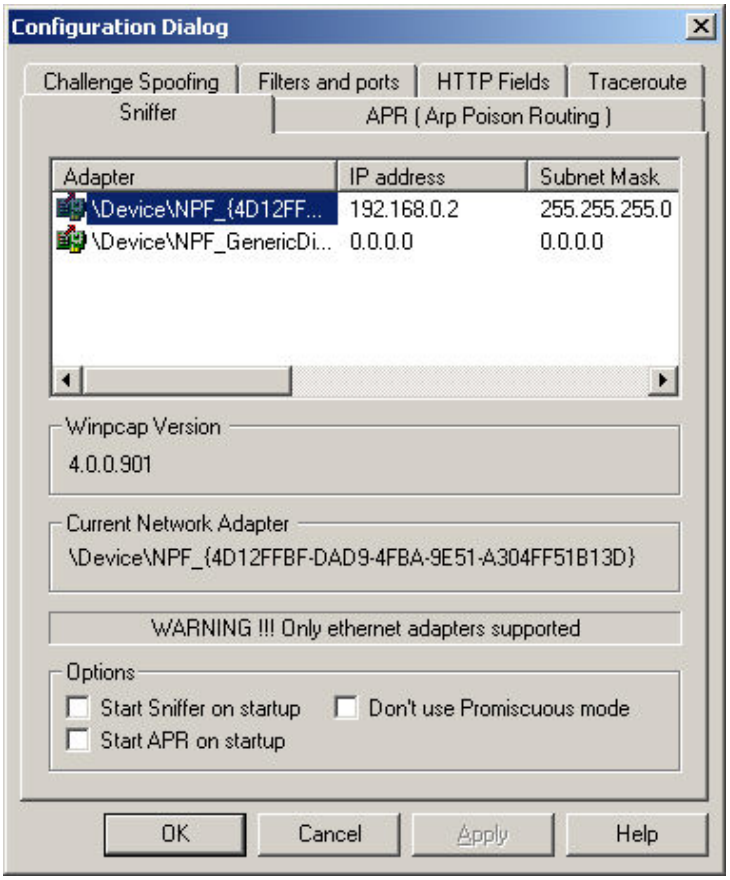
You can remove the Abel Service using Cain's Service Manager; first stop the service and then remove it.

Configuration

Cain & Abel requires the configuration of some parameters; everything can be set from the main configuration dialog.

Sniffer Tab

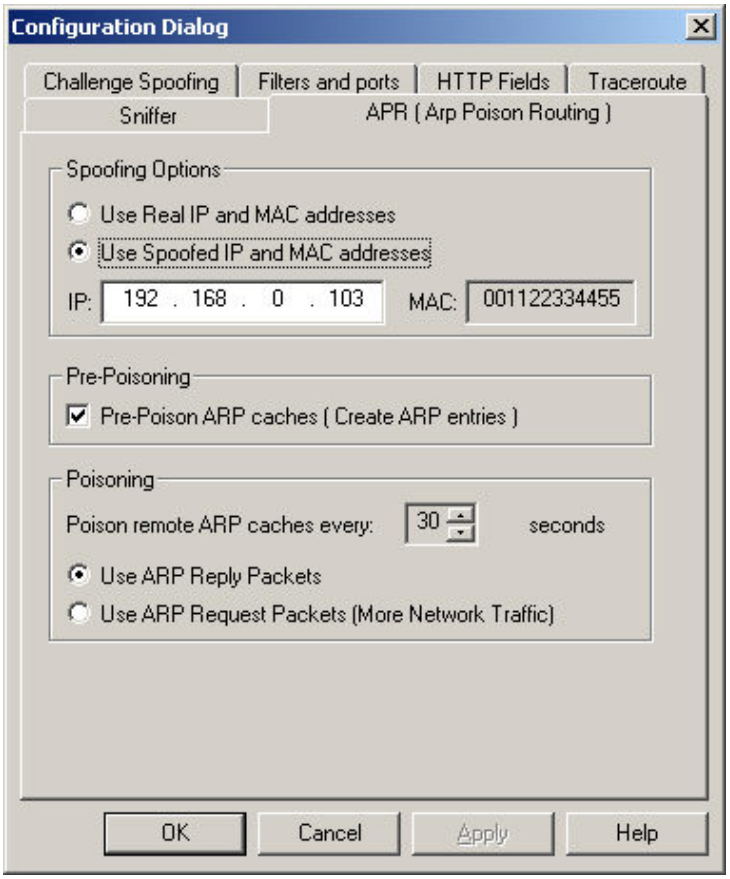
Here you can set the network card to be used by Cain's sniffer and APR features. The last two check boxes enable/disables these functions at the program's startup.
The sniffer is compatible with Winpcap drivers of version 2.3 or later and in this version only Ethernet adapters are supported by the program.



If enabled, the option "Don't use Promiscuous mode" enables APR Poisoning on wireless networks but please note that in this situation you cannot use the MAC spoofing feature below !

APR Tab

This is where you can configure APR (Arp Poison Routing). Cain uses a separate thread that sends ARP Poison packets to victim hosts every 30 seconds by default. This is necessary because entries present in the ARP cache of remote machines can be flushed out in case of no traffic. From this dialog you can set the time between each ARP Poison storm: setting this parameter to few seconds will cause a lot of ARP network traffic while setting it for long delays could not produce the desired traffic hijacking.



The spoofing options define the addresses that Cain writes into the Ethernet, ARP headers of ARP Poison Packets and re-routed packets. In this case the ARP Poison attack will be completely anonymous because the attacker's real MAC an IP addresses are never sent on the network.

If you want to enable this option you must consider that:

- Ethernet address spoofing can be used only if the attacker's workstation is connected to a HUB or to a network switch that does not use the "Port Security" feature. If "Port Security" is enabled on the switch, the source MAC address contained in every ethernet frame is checked against a list of allowed MAC addresses set on the switch. If the spoofing MAC address is not in this list the switch will disable the port and you will loose connectivity.

- The spoofing IP address must be a free address of your subnet. The ARP protocol does not cross routers or VLANs so if you set a spoofing IP that is out of your subnet the remote host will reply to it's default gateway and you will not see its responses. Also if you use a spoofing IP address that is already used in your subnet there will be an "IP address conflict" and the attack will be easily noticed. Here are some examples of valid spoofing addresses:

Real IP address	**Subnet Mask**	**Valid range for the spoofing IP address**
192.168.0.1	255.255.255.0	Must be an unused address in the range 192.168.0.2 - 192.168.0.254
10.0.0.1	255.255.0.0	Must be an unused address in the range 10.0.0.2 - 10.0.255.254
172.16.0.1	255.255.255.240	Must be an unused address in the range 172.16.0.2 - 172.16.0.14
200.200.200.1	255.255.255.252	Must be an unused address in the range 200.200.200.2 - 200.200.200.3

The spoofing IP address is automatically checked by the program when you press the "Apply" button, if the address is already in use in the subnet a message box will report the problem.

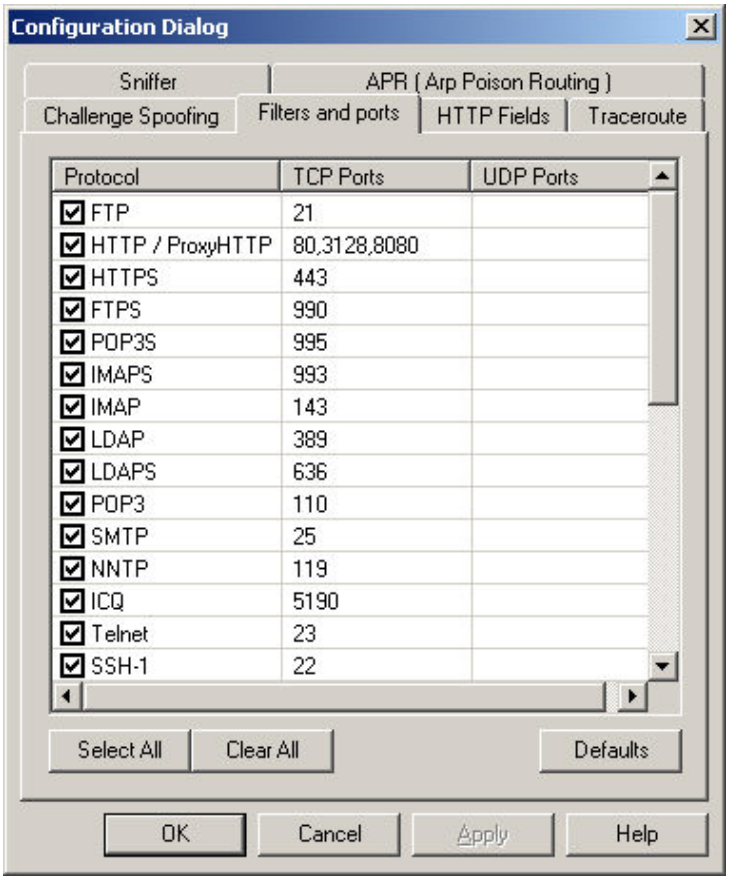
- The spoofing MAC address must not be present in your subnet. The presence of two identical MAC addresses on the same Layer-2 LAN can cause switches convergence problems; for this reason I decided to not let you easily set the spoofing MAC of your choice from the configuration dialog. The default value is set to 001122334455 which is an invalid address not supposed to exist in your network and that at the same time can be easily identified for troubleshooting. **IMPORTANT ! You cannot have, on the same Layer-2 network, two or more Cain machines using APR's MAC spoofing and the same Spoofed MAC address**. The spoofing MAC address can be changed modifying the registry value "SpoofMAC" at this location: "HKEY_CURRENT_USER\Software\Cain\Settings".

Filters and Ports Tab

Here you can enable/disable Cain's sniffer filters and application protocol TCP/UDP ports. Cain captures only authentication

information not the entire content of each packet, however you can use the Telnet filter to dump, into a file, all the data present in a TCP session, modifying the relative filter port.

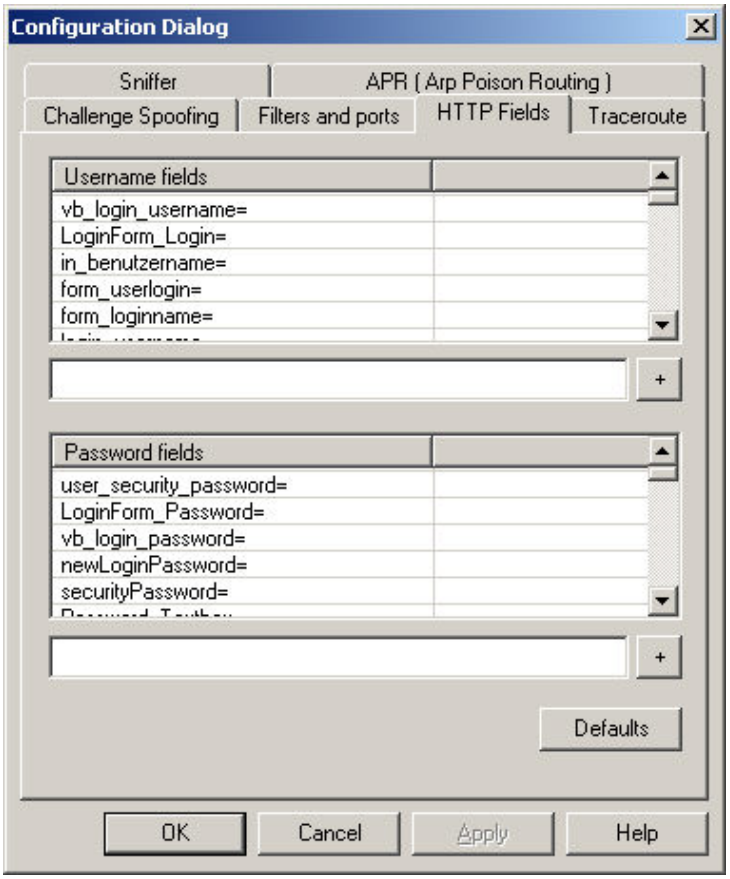
Cain's sniffer filters are internally designed to survive in an unreliable world such as a network under ARP Poison attack; Cain uses different state machines to extract from network packets all the information needed to recover the plaintext form of a transmitted password. Some authentication protocols uses a challenge-response mechanism so it needs to collect parameters from Client->Server and Server->Client traffic; traffic interception in both directions is always possible if your Level-2 network is made by HUBs only or if you are connected to a mirror port on the switch but on switched networks in general, it can be achieved only using some kind of traffic hijacking technique such as Arp Poison Routing (APR). If you are sniffing with APR enabled, the sniffer will extract challenge-response authentications only if you reach a Full-Routing state between victim computers.



Under this tab you can also enable/disable the analysis of routing protocols (HSRP, VRRP, EIGRP, OSPF, RIPv1, RIPv2) and the APR-DNS feature that acts as a DNS Reply Rewriter.

HTTP Fields Tab

This tab contains a list of user name and password fields to be used by the HTTP sniffer filter. Cookies and HTML Forms that travel in HTTP packets are examined in this way: for each user name field all the password fields are checked and if these two parameters are found, the credentials will be captured and displayed on the screen.



The following cookie uses the fields "logonusername=" and "userpassword=" for authentication purposes; if you don't include these two fields in the above list the sniffer will not extract relative credentials.

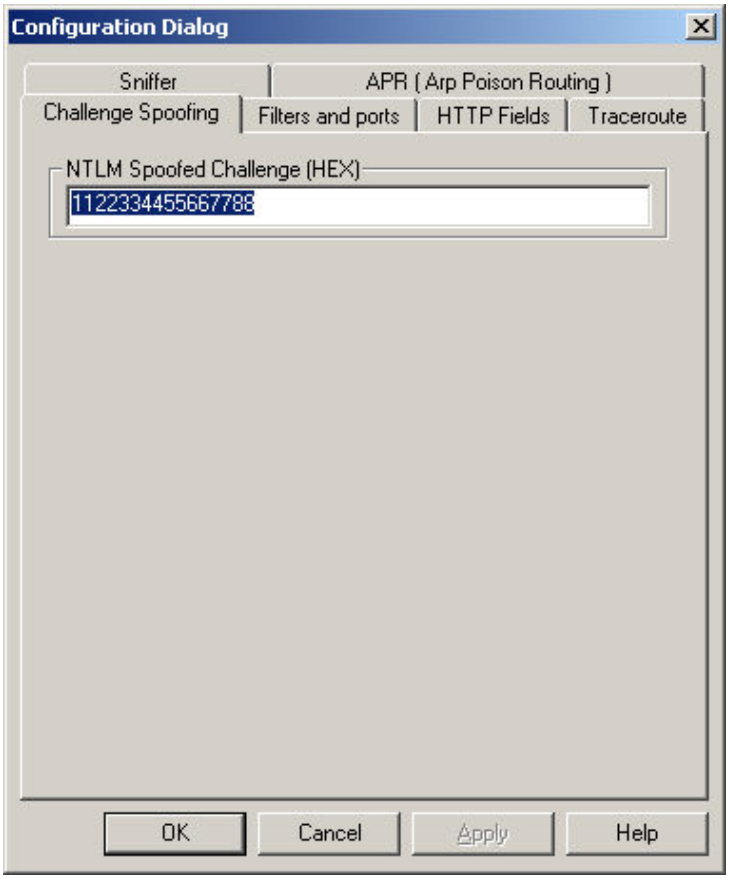
GET /mail/Login?domain=xxxxxx.xx&style=default&plain=0 HTTP/1.1
*Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */**
Referer: http://xxx.xxxxxxx.xx/xxxxx/xxxx
Accept-Language: it
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; (R1 1.3); .NET CLR 1.1.4322)
Host: xxx.xxxxxx.xx
Connection: Keep-Alive
Cookie: ss=1; logonusername=user@xxxxxx.xx; ss=1; srclng=it; srcdmn=it; srctrg=_blank; srcbld=y; srcauto=on; srcclp=on; srcsct=web; userpassword=password; video=c1; TEMPLATE=default;

Traceroute Tab

This is used to configure Cain's ICMP/UDP/TCP traceroute. You can set to resolve host names, use ICMP Mask discovery and enable/disable WHOIS information extraction for each hop.

Challenge Spoofing Tab

Here you can set the custom challenge value to rewrite into NTLM authentications packets. This feature can be enabled quickly from Cain's toolbar and must be used with APR. A fixed challenge enables cracking of NTLM hashes captured on the network by mean of RainbowTables.



Features overview

Cain's features

- Protected Storage Password Manager

Reveals locally stored passwords of Outlook, Outlook Express, Outlook Express Identities, Outlook 2002, Internet Explorer and MSN Explorer.

- Credential Manager Password Decoder

Reveals passwords stored in Enterprise and Local Credential Sets on Windows XP/2003.

- LSA Secrets Dumper

Dumps the contents of the Local Security Authority Secrets.

- Dialup Password Decoder

Reveals passwords stored by Windows "Dial-Up Networking" component.

- APR (ARP Poison Routing)

Enables sniffing on switched networks and Man-in-the-Middle attacks.

- Route Table Manager

Provides the same functionality of the Windows tool "route.exe" with a GUI front-end.

- SID Scanner

Extracts user names associated to Security Identifiers (SIDs) on a remote system.

- Network Enumerator

Retrieves, where possible, the user names, groups, shares, and services running on a machine.

- Remote Registry

Allows modification of registry parameters from the network.

- Service Manager

Allows you to stop, start, pause/continue or remove a service.

- Sniffer

Captures passwords, hashes and authentication information while they are transmitted on the network. Includes several filters for application specific authentications and routing protocols. The VoIP filter enables the capture of voice conversations transmitted with the SIP/RTP protocol saved later as WAV files.

- Routing Protocol Monitors

Monitors messages from various routing protocols (HSRP, VRRP, RIPv1, RIPv2, EIGRP, OSPF) to capture authentications and shared route tables.

- Full RDP sessions sniffer for APR (APR-RDP)

Allows you to capture all data sent in a Remote Desktop Protocol (RDP) session on the network. Provides interception of keystrokes activity client-side.

- Full SSH-1 sessions sniffer for APR (APR-SSH-1)

Allows you to capture all data sent in SSH-1 sessions on the network.

- Full HTTPS sessions sniffer for APR (APR-HTTPS)

Allows you to capture all data sent in HTTPS sessions on the network.

- Full FTPS sessions sniffer for APR (APR-FTPS)

Allows you to capture all data sent in implicit FTPS sessions on the network.

- Full POP3S sessions sniffer for APR (APR-POP3S)

Allows you to capture all data sent in implicit POP3S sessions on the network.

- Full IMAPS sessions sniffer for APR (APR-IMAPS)

Allows you to capture all data sent in implicit IMAPS sessions on the network.

- Full LDAPS sessions sniffer for APR (APR-LDAPS)

Allows you to capture all data sent in implicit LDAPS sessions on the network.

- Certificates Collector

Grab certificates from HTTPS, IMAPS, POP3S, LDAPS, FTPS web sites and prepares them to be used by relative APR-* sniffer filters.

- MAC Address Scanner with OUI fingerprint

Using OUI fingerprint, this makes an informed guess about what type of device the MAC address from.

- Promiscuous-mode Scanner based on ARP packets

Identifies sniffers and network Intrusion Detection systems present on the LAN.

- Wireless Scanner

Can scan for wireless networks signal within range, giving details on its MAC address, when it was last seen, the guessed vendor, signal strength, the name of the network (SSID), whether it has WEP or not (note WPA encrypted networks will show up as WEPed), whether the network is an Ad-Hoc network or Infrastructure, what channel the network is operating at and at what speed the network is operating (e.g. 11Mbps). Passive scanning and WEP IVs sniffing are also supported using the AirpCap adapter from CACE Technologies.

- 802.11 Capture Files Decoder

Decode 802.11 capture files (wireshark, pcap) containing wireless frames encrypted with WEP or WPA-PSK.

- Access (9x/2000/XP) Database Passwords Decoder

Decodes the stored encrypted passwords for Microsoft Access Database files.

- Base64 Password Decoder

Decodes Base64 encoded strings.

- Cisco Type-7 Password Decoder

Decodes Cisco Type-7 passwords used in router and switches configuration files.

- Cisco VPN Client Password Decoder

Decodes Cisco VPN Client passwords stored in connection profiles (*.pcf).

- VNC Password Decoder

Decodes encrypted VNC passwords from the registry.

- Enterprise Manager Password Decoder

Decodes passwords used by Microsoft SQL Server Enterprise Manager (SQL 7.0 and 2000 supported).

- Remote Desktop Password Decoder

Decodes passwords in Remote Desktop Profiles (.RPD files).

- PWL Cached Password Decoder

Allows you to view all cached resources and relative passwords in clear text either from locked or unlocked password list files.

- Password Crackers

Enables the recovery of clear text passwords scrambled using several hashing or encryption algorithms. All crackers support Dictionary and Brute-Force attacks.

- Cryptanalysis attacks

Enables password cracking using the 'Faster Cryptanalytic time – memory trade off' method introduced by Philippe Oechslin. This cracking technique uses a set of large tables of pre calculated encrypted passwords, called Rainbow Tables, to improve the trade-off methods known today and to speed up the recovery of clear text passwords.

- WEP Cracker

Performs Korek's and PTW WEP attacks on 802.11 capture files containing enough WEP initialization vectors.

- Rainbowcrack-online client

Enables password cracking by mean of the outstanding power of this on-line cracking service based on RainbowTable technology.

- NT Hash Dumper + Password History Hases (works with Syskey enabled)

Will retrieve the NT password hash from the SAM file regardless of whether Syskey in enabled or not.

- Syskey Decoder

Will retrieve the Boot Key used by the SYSKEY utility from the local registry or "off-line" SYSTEM files.

- MSCACHE Hashes Dumper

Will retrieve the MSCACHE password hashes stored into the local registry.

- Wireless Zero Configuration Password Dumper

Will retrieve the wireless keys stored by Windows Wireless Configuration Service.

- Microsoft SQL Server 2000 Password Extractor via ODBC

Connects to an SQL server via ODBC and extracts all users and passwords from the master database.

- Oracle Password Extractor via ODBC

Connects to an Oracle server via ODBC and extracts all users and passwords from the database.

- MySQL Password Extractor via ODBC

Connects to an MySQL server via ODBC and extracts all users and passwords from the database.

- Box Revealer

Shows passwords hidden behind asterisks in password dialog boxes.

- RSA SecurID Token Calculator

Can calculate the RSA key given the token's .XML activation file.

- Hash Calculator

Produces the hash values of a given text.

- TCP/UDP Table Viewer

Shows the state of local ports (like netstat).

- TCP/UDP/ICMP Traceroute with DNS resolver and WHOIS client

A improved traceroute that can use TCP, UDP and ICMP protocols and provides whois client capabilities.

- Cisco Config Downloader/Uploader (SNMP/TFTP)

Downloads or uploads the configuration file from/to a specified Cisco device (IP or host name) given the SNMP read/write community string.

Abel features

- Remote Console

Provides a remote system shell on the remote machine.

- Remote Route Table Manager

Enable to manage the route table of the remote system.

- Remote TCP/UDP Table Viewer

Shows the state of local ports (like netstat) on the remote system.

- Remote NT Hash Dumper + Password History Hases (works with Syskey enabled)

Will retrieve the NT password hash from the SAM file regardless of whether Syskey in enabled or not; works on the Abel-side.

- Remote LSA Secrets Dumper

Dumps the contents of the Local Security Authority Secrets present on the remote system.

Abel

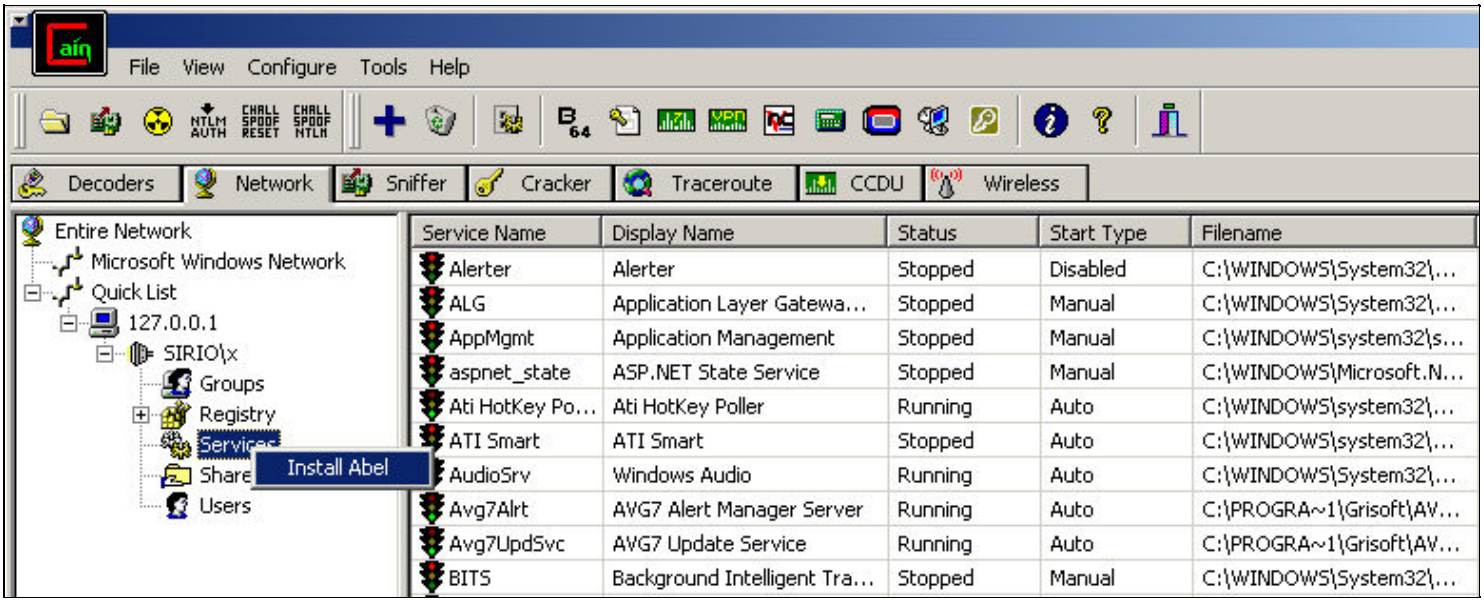
Abel is the second part of the program. Designed as a Windows NT service it is composed by two files "Abel.exe" and "Abel.dll"; the first is the main service executable program and the second is a library that contains some required functions. Although Cain is the main Abel's front-end, it is not needed to be installed for Abel to work.
The service can be installed locally or remotely (using Cain) and requires Administrator's privileges on the target machine.

Local Installation

Abel's local installation is really simple; all you have to do is to copy the executable files in a directory and type "Abel" at the command prompt.

Remote Installation

The remote installation is even more simple. Cain's Service Manager will do everything for you from the file copy to the service creation.



Once installed, as all other NT services, it can be managed using the standard Windows tools or the Cain's Service Manager.

Abel communicates with Cain using a the Windows named pipe "\\computername\pipe\abel" and it can accept connections from multiple hosts at the same time. All data transmitted over this pipe is encrypted using the RC4 symmetric encryption algorithm and

the fixed key "Cain & Abel". This is done only to scramble the traffic sent on the network and not to hide program's intentions.

The service runs using the Local System account and provides some interesting features like the Remote NT Hashes Dumper, the Remote LSA Secrets Dumper and the Remote Console.

How to remove Abel

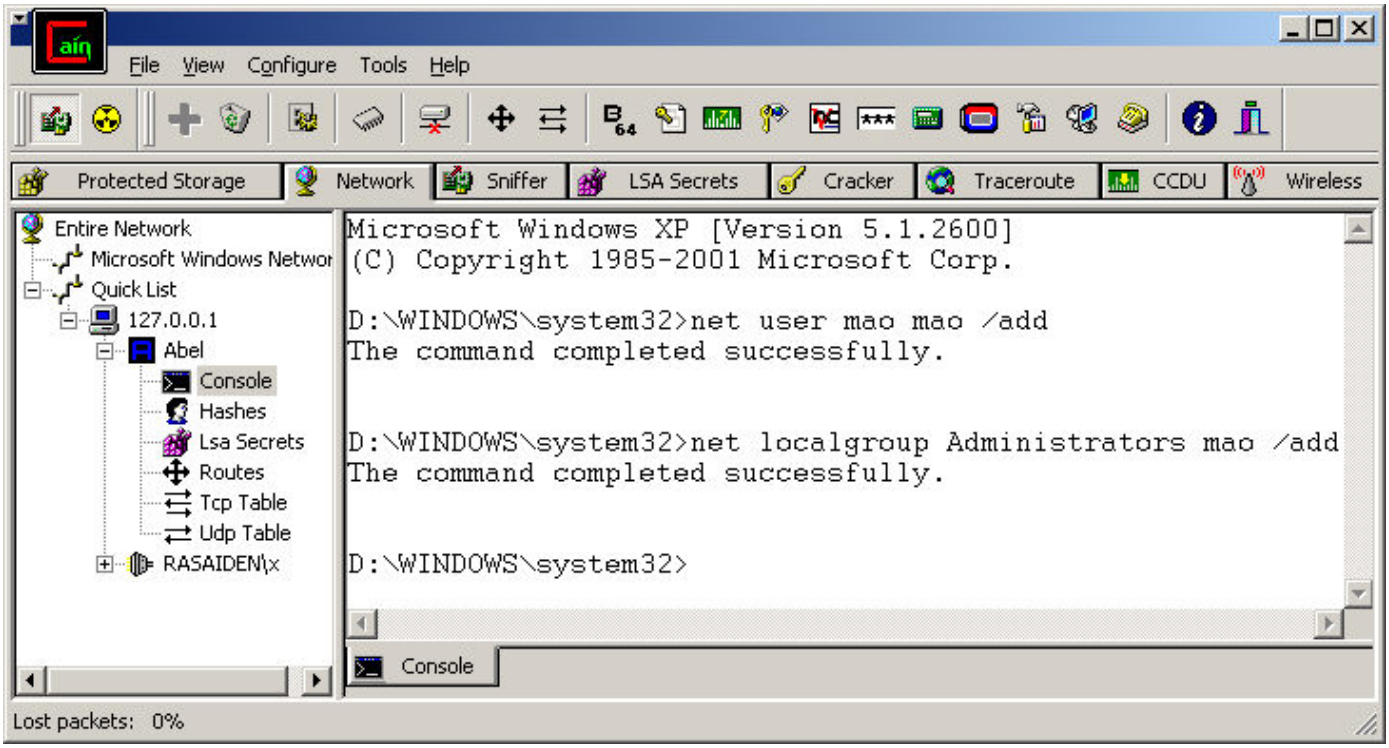
Stop the service and then type "Abel -r" at the command prompt. You can also use the Cain's Service Manager to do that. Once the service is removed the executable files can be manually deleted from the system.

Usage

Some Security Software vendors like Internet Security Systems (ISS) classified Abel as an High Risk backdoor program (http://xforce.iss.net/xforce/xfdb/17320) and now they have signatures to recognize the program. Although I am not interested in their decision, I suggest to you to use Abel for educational purposes only.

Remote Console

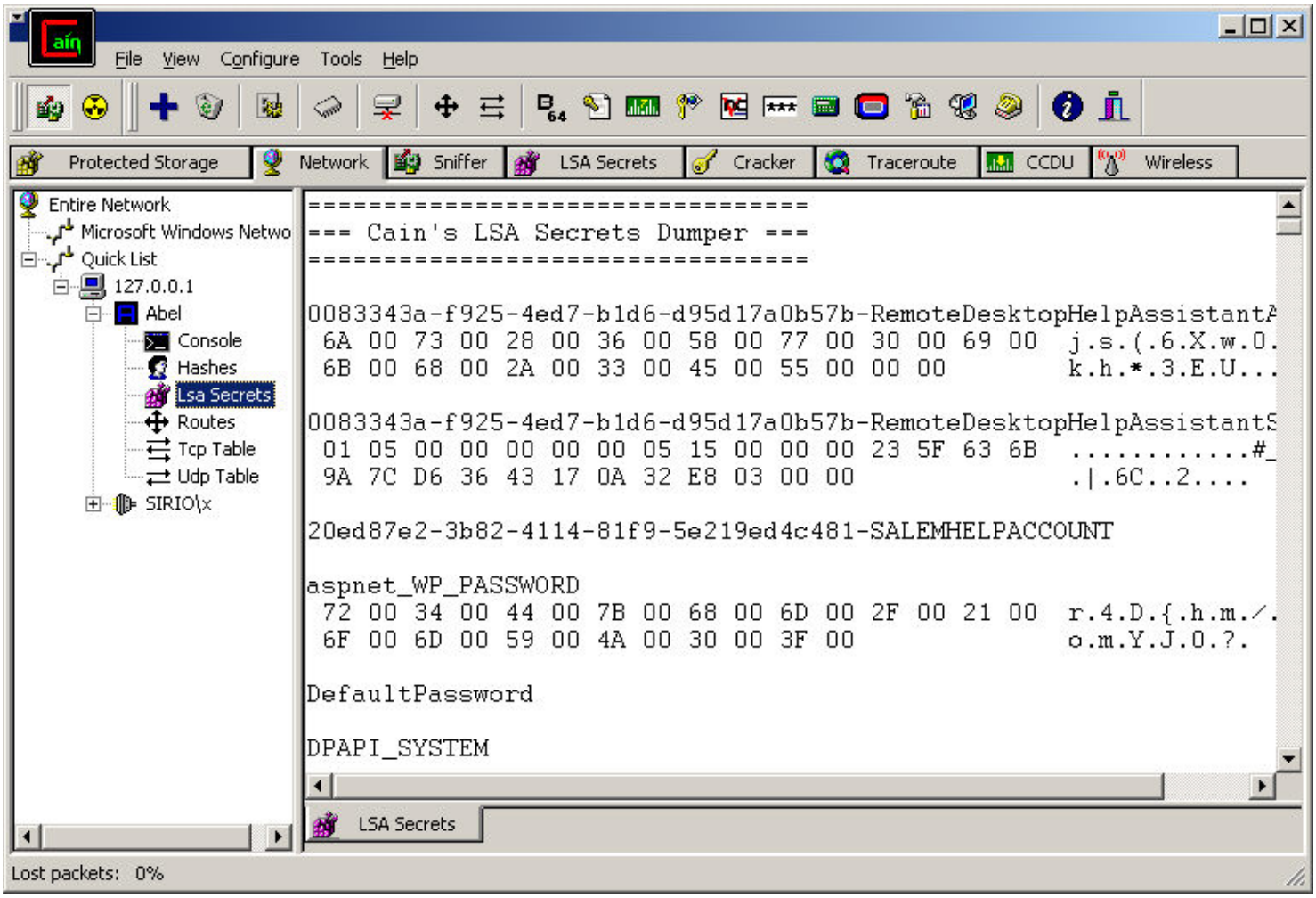
Abel's remote console provides a system shell on the remote machine. The Abel service runs on the remote machine in the security context of it's Local System Account; every command sent to the console is executed with the same access privileges of that account.



You can use the Abel console to add users, enumerate networks, ping remote hosts, map network drives and so on .... every command is executed on the Abel-side.

Remote LSA Secrets Dumper

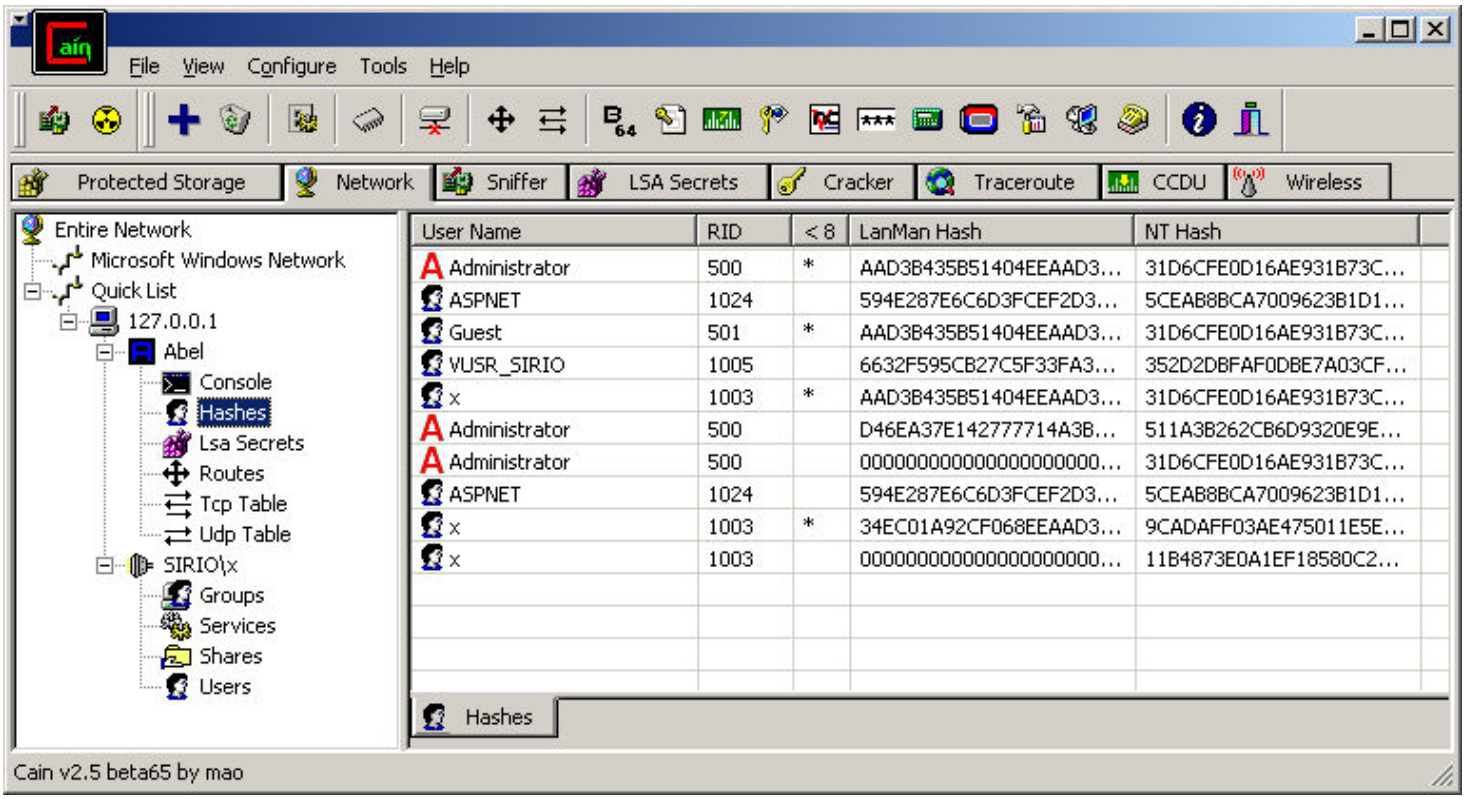
Provides the same functionality of Cain's LSA Secrets Dumper but on the Abel-side.



Abel service runs using the Local System Account the has, by default, all the required privileges to dump LSA Secrets.

Remote NT Hashes Dumper

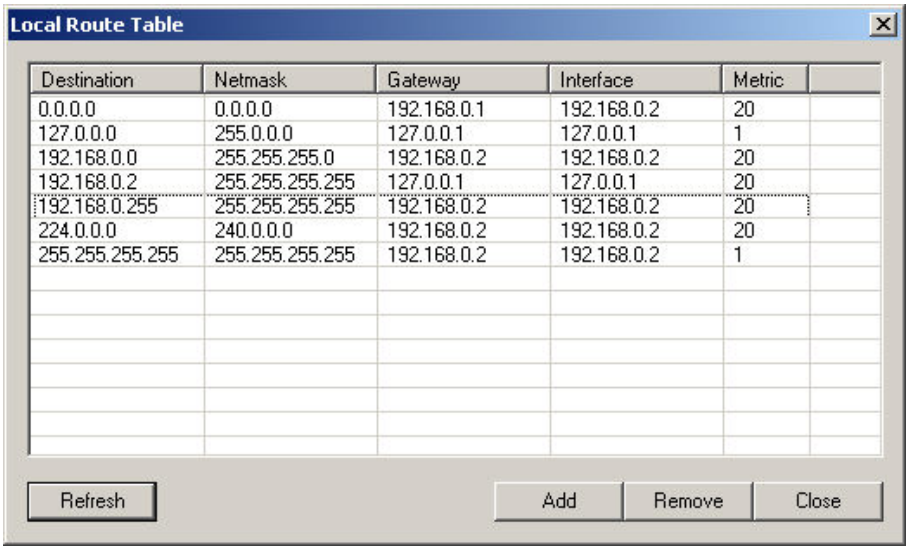
Provides the same functionality of Cain's NT Hashes Dumper but on the Abel-side.

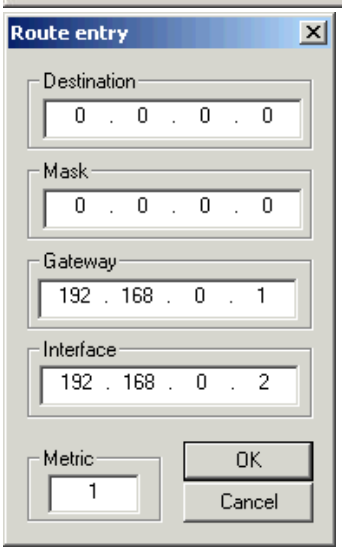


Abel service runs using the Local System Account the has, by default, all the required privileges to dump password hashes. Once dumped the hashes can be sent to Cain's LM & NTLM cracker for the clear text password recovery.

Route Table Manager

Cain's Route Table manager covers the same functionality offered by the Windows tool "route.exe".





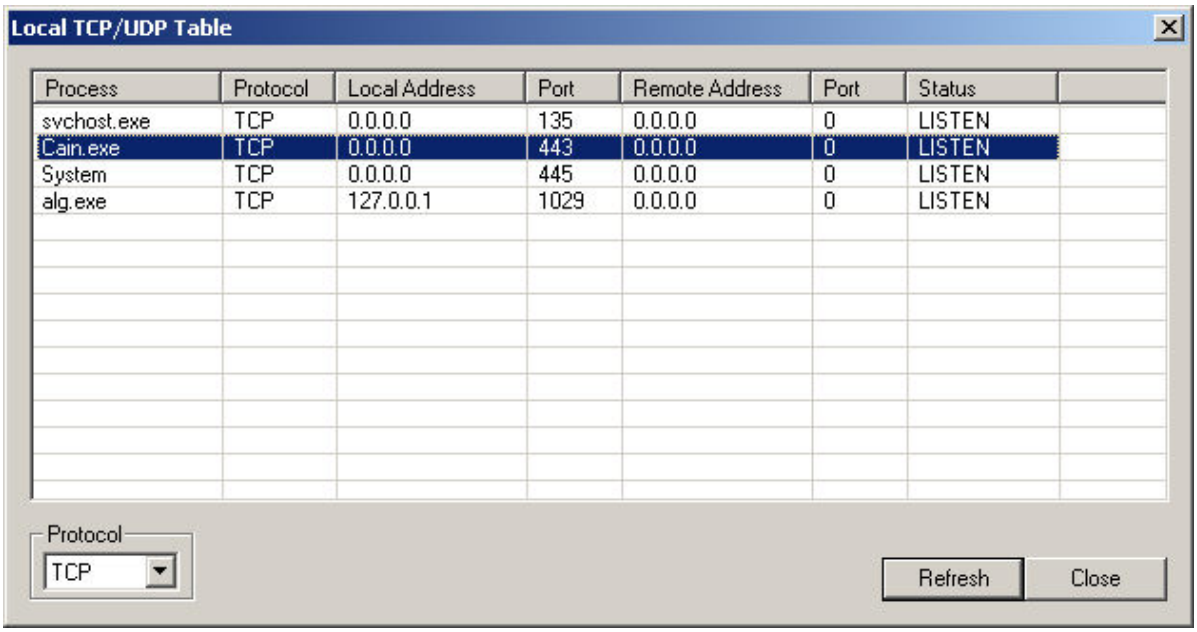
This utility enumerates all entries present in the local Windows route table. It also enables you to add new routes and modify/remove existing ones.

Usage

The utility can be activated using the relative toolbar button.

TCP/UDP Table Viewer

Cain's TCP/UDP Table viewer covers the same functionality offered by the Windows tool "netstat.exe".



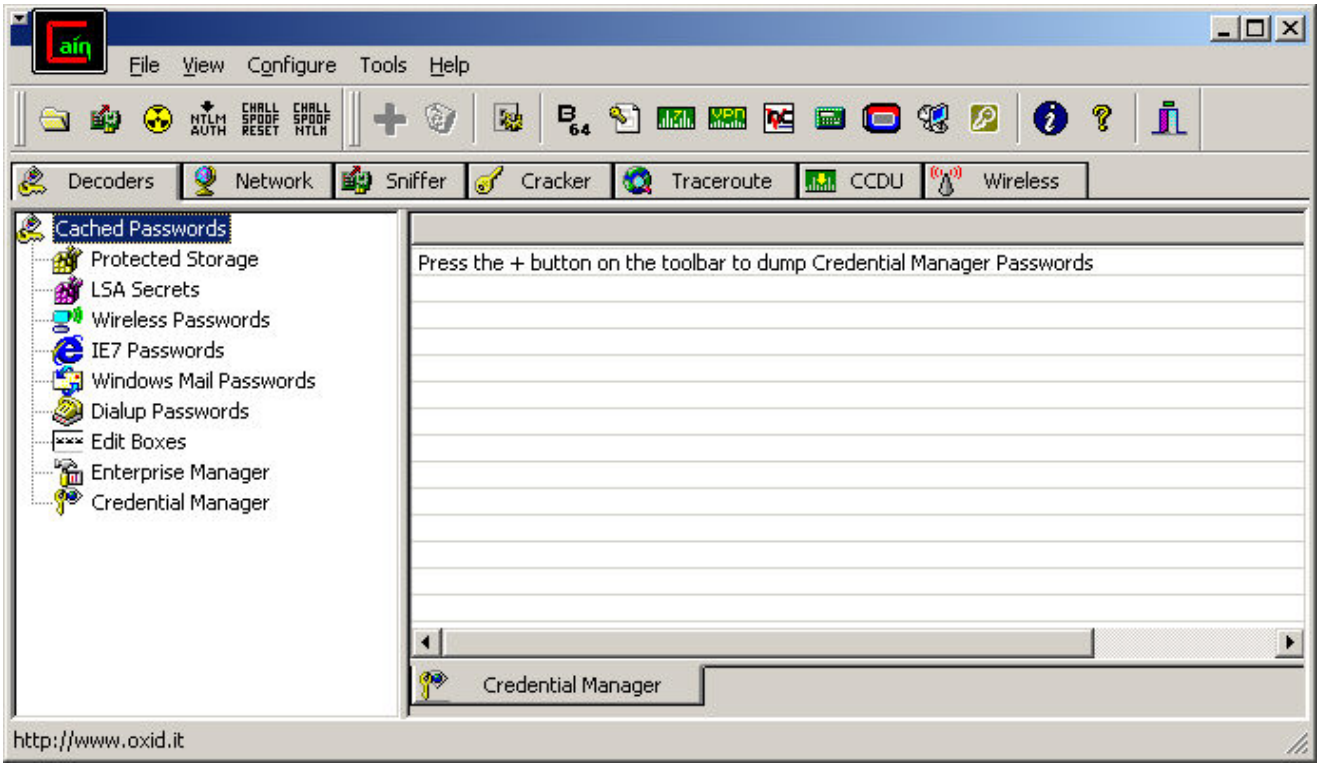
The dialog shows all connection's parameters as protocol type, local address, port number, remote address and connection's status (TCP only). On Windows XP, the filename of the process associated to TCP and UDP ports is also displayed.

Note

When executed, Cain binds a TCP socket on port 443 on the local machine. This is needed by the APR-HTTPS feature present in the program; Cain does not open any other local port in listening mode. For more information please refer to this page.

Cain

Cain is the first part of the program. Developed with a simple Windows graphical user interface, its main purpose is to concentrate several hacking techniques and proof of concepts providing a simplified tool focused on the recovery of passwords and authentication credentials from various sources.



The user interface is made up of lists, tabs, toolbars and dialogs. Lists are used to contain similar kind of information such as passwords, host names, hashes and other parameters while tabs are used to switch from features. The sniffer toolbar contains only two buttons to activate/deactivate the sniffer and APR functions and the main toolbar the remaining commands to interact with the program.

Note

The blue "+" and the recycle bin buttons on the main toolbar have the general mean to "add something to" and "remove something from" the current list but they performs different actions depending on the currently focused list.

There are also some general Hot Keys in the program to accelerate some functions:

Insert -> Add to List, Insert new entry, new item .....
Delete -> Remove selected items from the current list, Remove all .....
Alt+Del -> Hide Cain's main window
Alt+PgDwn -> Send Cain's main window to system tray
Alt+PgUp -> Restore Cain's main window
F5 -> Refresh list contents

Cain is compiled using Microsoft Visual C++ and linked as a single executable file named "Cain.exe". Although it is principally written in C/C++ a lot of code has been optimized in Assembler for performance reasons. MFC classes and OpenSSL libraries are statically linked to the program; this increases the file size but eliminates the need to distribute shared DLLs.

Cisco Config Downloader/Uploader

This feature allows you to download or upload the configuration file of Cisco devices via SNMP/TFTP. It supports routers and switches that uses the OLD-CISCO-SYSTEM-MIB or the new CISCO-CONFIG-COPY-MIB; for more information about those MIBs please refer to Cisco web site.

How it works

1) Cain requests the configuration file transfer to the Cisco device using the SNMP protocol. Request packets are constructed using some proprietary Cisco OIDs that the vendor provides for this functionality; they also contains other parameters like the protocol type, the server IP address and filenames to instruct the device on where to send or to take its configuration file.
2) At this point the device starts the file transfer using the protocol specified in the request (set to TFTP for simplicity).
3) Cain opens a TFTP socket in listening mode and handles the file transfer. A TFTP server is NOT required, when uploading the program sends the configuration file to the device, when downloading it receives it.

Usage

To download a configuration from a device press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar, provide the IP address of the SNMP enabled device and the right Read/Write Community string. To upload a configuration use the relative function within the list pop up menu.

Limitations

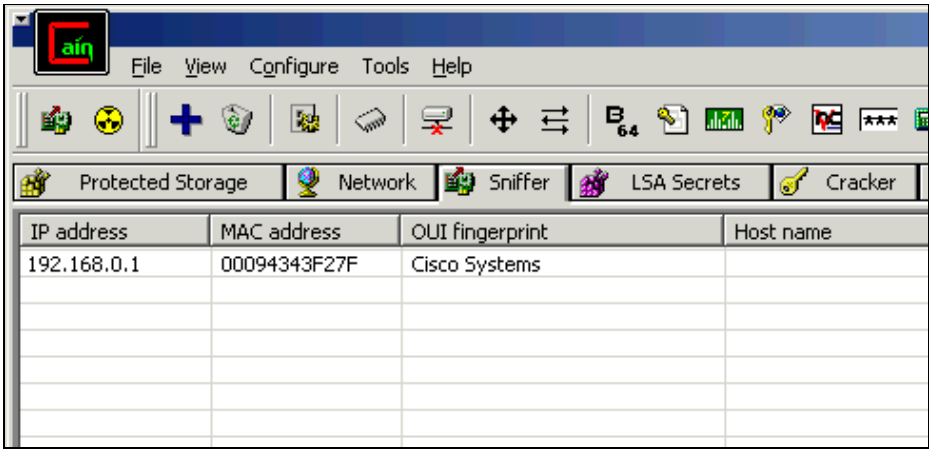
This feature will not work if network restrictions, like ACLs or firewall rules, for interested protocols (SNMP/TFTP) are set. The TFTP file transfer is initiated by the device itself so dynamic NAT between you and the device is a problem as well.

Requirements

- CCDU works on Cisco Routers and Switches that supports the OLD-CISCO-SYSTEM-MIB or the new CISCO-CONFIG-COPY-MIB. PIX Firewalls does not support those MIBs.
- You also need the right Read/Write SNMP community string (e.g.: "private"), the Read-Only one is not enough.

MAC Scanner

The MAC address scanner is a very fast IP to MAC address resolver based on ARP Request/Reply packets. It takes as input a range of IP addresses on the current subnet and resolves the MAC addresses associated to those IP's. The scanner includes an OUI database, providing MAC vendor's information, this feature is useful to quickly identify switches, routers, load balancers and firewalls present in the LAN.



Because of the use of ARP packets that cannot cross routers or VLANs, this feature can resolve MAC addresses in the local broadcast domain only. The OUI database is a normalized version of the IEEE OUI list available at this link: http://standards.ieee.org/regauth/oui/index.shtml.
Once active hosts are found, you can also resolve their host names with the "Resolve Host Name" function within the list pop up menu.

Tip

The scanner cannot resolve MAC addresses if the network card is not correctly configured. You also have to check the APR's spoofing options in the configuration dialog before initiating a scan.

Prerequisites

The sniffer must be activated.

Usage

The scanner's configuration dialog is activated pressing the "Insert" button on the keyboard or click the icon with the blue + on the toolbar; then you have to select the range of IP addresses to resolve.

Network Enumerator

The Network Enumerator uses the native Windows network management functions (Net*) to discover what is present on the network. It allows a quick identification of Domain Controllers, SQL Servers, Printer Servers, Remote Access Dial-In Servers, Novell Servers, Apple File Servers, Terminal Servers and so on. It can also display when possible the version of their operating system.

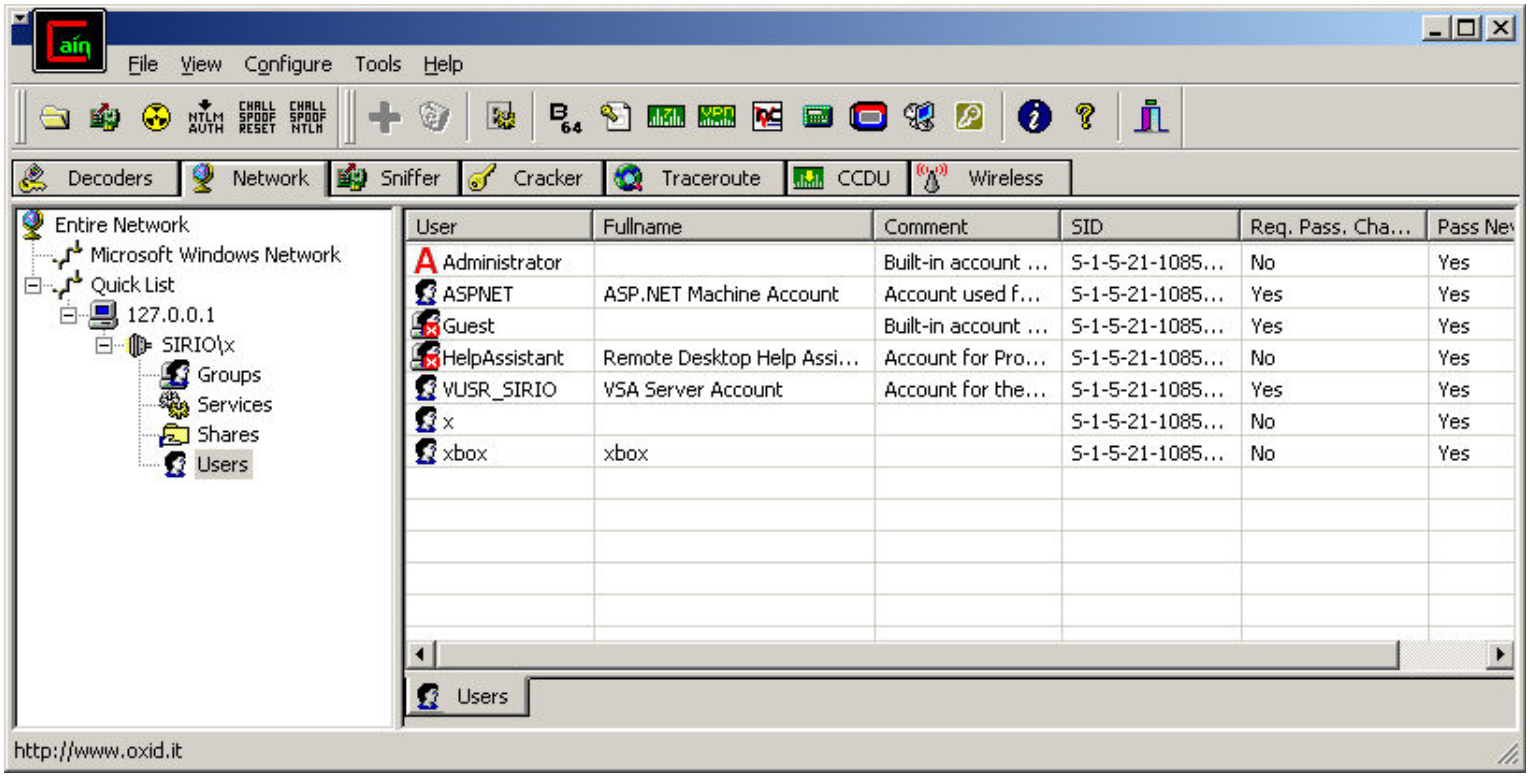
The left tree is used to browse the network and to connect to remote machines; once connected to a server you can also enumerate user names, groups, services and shares present on it. By default the program connects to remote IPC$ shares using the current local logged on user and if it fails using NULL sessions (Anonymous sessions); however it is also possible to specify the credentials to be used for the connection. The Quick List can be used to insert IP addresses of hosts that aren't seen browsing the network.

When enumerating users, Cain also extracts their Security Identifier (SID) and has the ability to identify the name of the Administrator account even if it was renamed. This is done by looking at the account RID which is the last part of a SID. The RID of the Administrator account is always equal to 500.

Windows NT and later has a security feature that can restrict the ability for anonymous logon users (also known as NULL session connections) to list account names and enumerate share names. This is done setting to 1 the parameter "**RestrictAnonymous**" under the registry key:

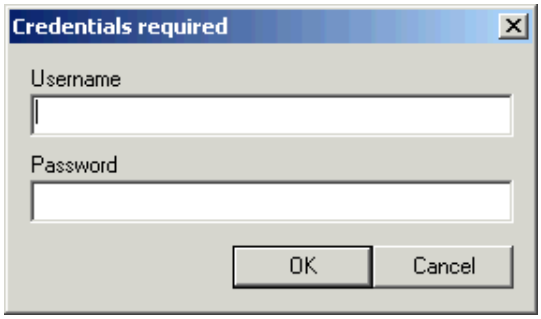
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA

If the program cannot enumerate users, because of this restriction, it will start automatically the SID Scanner and will proceed with an extraction of them using the same methodology used by the well known tool **sid2user** by Evgenii B. Rudnyi.



Tip

To perform an Anonymous connection (NULL Session) to the target host, leave the user name and password fields empty in the credentials dialog.

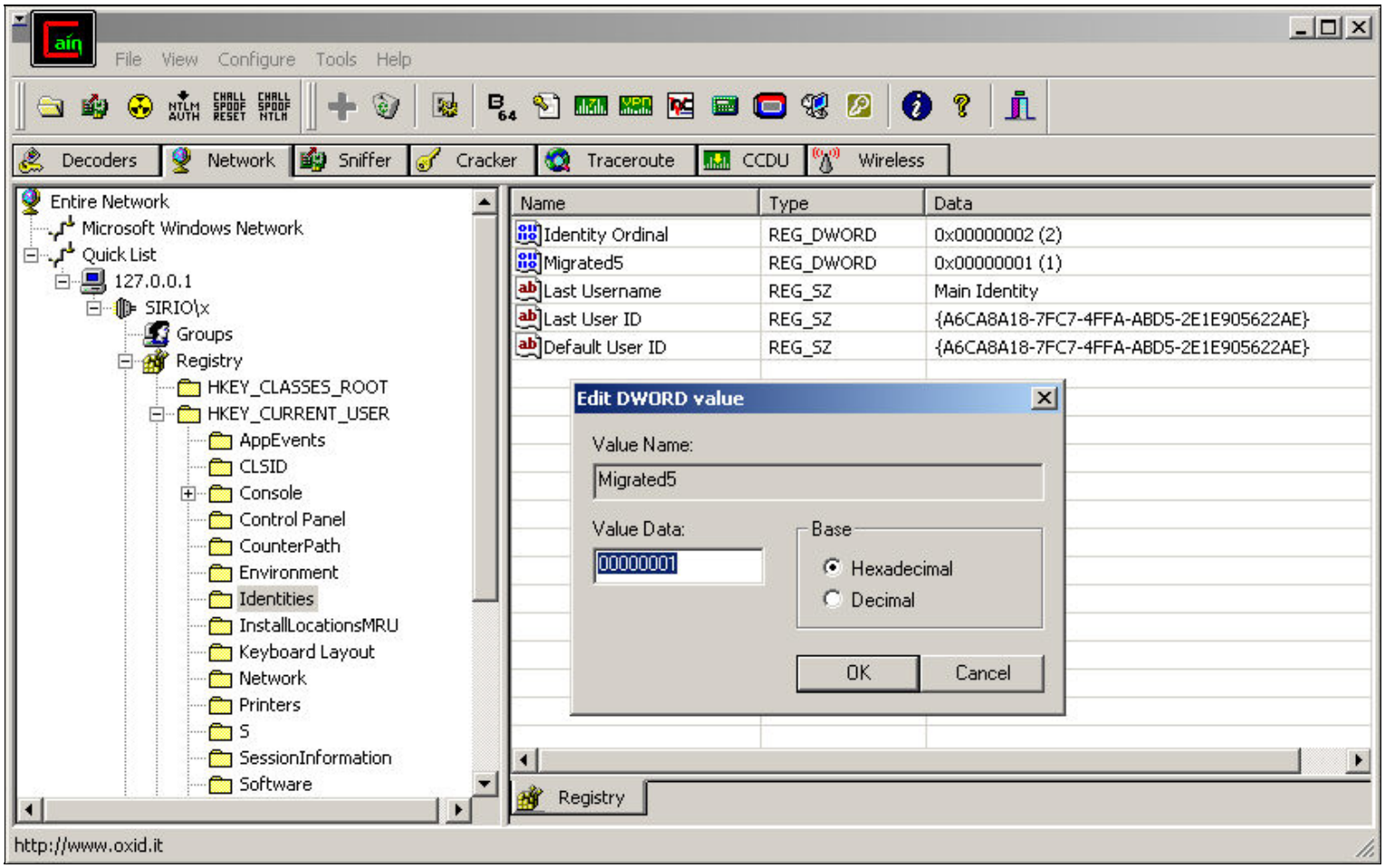


Usage

Enumerations are launched browsing the tree on the left into the Network tab. To specify credentials for a network connection you can right click on the target machine and use the "Connect As" function within the pop up menu.

Remote Registry

The remote registry feature enables the manipulation of registry parameters from the network.



Requirements

This feature requires the "Remote Registry Service" running on the remote machine.

Usage

Enumerations are launched browsing the tree on the left within the Network tab. To specify credentials for a network connection you can right click on the target machine and use the "Connect As" function within the pop up menu.

Password Crackers

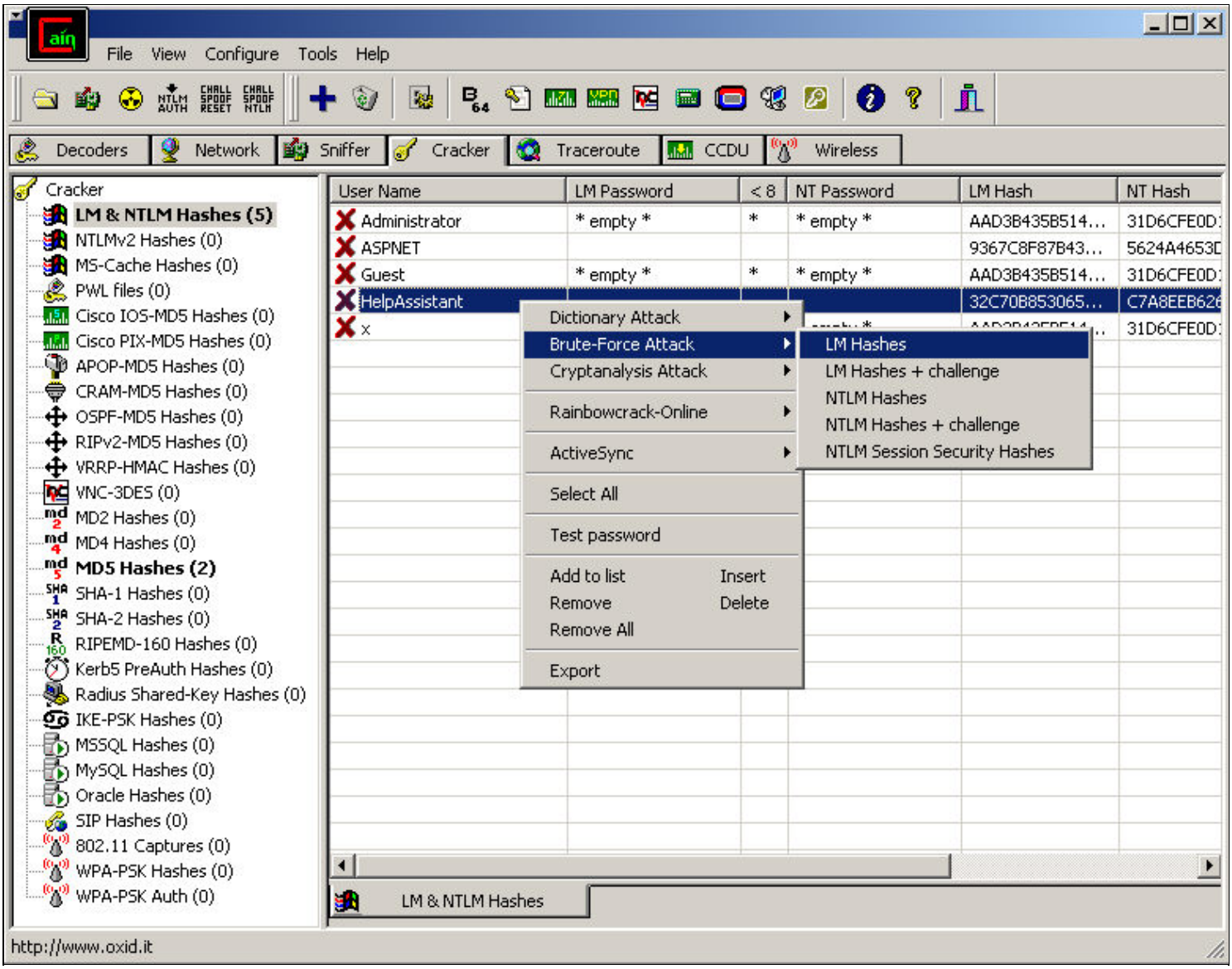
Cain's Password Crackers support the most common hashing algorithms and several encryption methods based on them :

Hash Types:

MD2, MD4, MD5, SHA1, SHA2 (256 bit), SHA2 (384 bit), SHA2 (512 bit), RIPEMD160.

Encryption algorithms:

PWL files, Cisco-IOS Type-5 enable passwords, Cisco PIX enable passwords, APOP-MD5, CRAM-MD5, LM, LM + Challenge, NTLM, NTLM + Challenge, NTLM Session Security, NTLMv2, RIPv2-MD5, OSPF-MD5, VRRP-HMAC-96, VNC-3DES, MS-Kerberos5 Pre-Auth, RADIUS Shared Secrets, IKE Pre-Shared Keys, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Oracle, Oracle-TNS-DES, Oracle-TNS-3DES, Oracle-TNS-AES128, Oracle-TNS-AES192, MySQL323, MySQLSHA1, SIP-MD5, WPA-PSK, WPA-PSK-AUTH, CHAP-MD5, MS-CHAPv1, MS-CHAPv2.



Password Crackers can be found in the program under the "Cracker" sub tab. The tree on the left allows you to select the list containing desired encrypted passwords or hashes to crack. Those selected are then loaded into the Dictionary / Brute-Force dialog using the relative function within the list pop up menu.

Brute-Force Password Cracker

A Brute-Force attack is method of breaking a cipher (that is, to decrypt a specific encrypted text) by trying every possible key. Feasibility of brute force attack depends on the key length of the cipher, and on the amount of computational power available to the attacker. Cain's Brute-Force Password Cracker tests all the possible combinations of characters in a pre-defined or custom character set against the encrypted passwords loaded in the brute-force dialog.

The key space of all possible combination of passwords to try is calculated using the following formula:

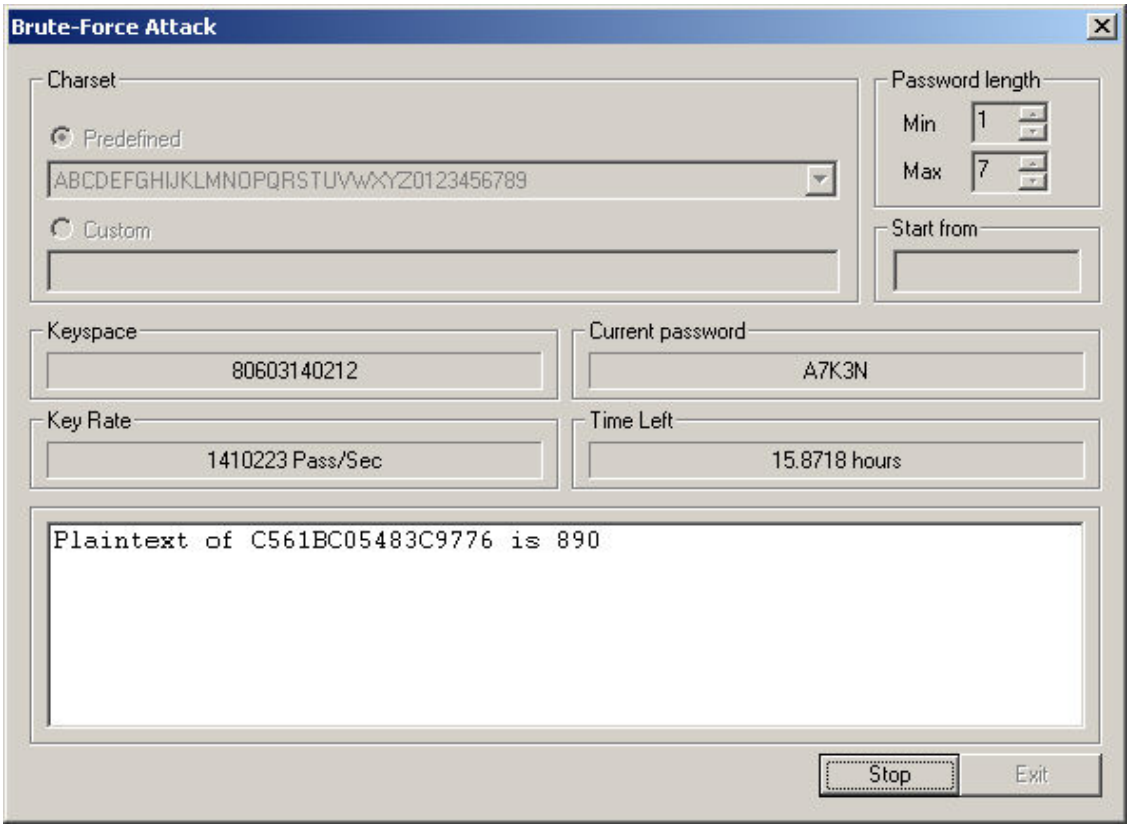
KS = L^(m) + L^(m+1) + L^(m+2) + ... + L^(M)

where

L = character set length
m = min length of the key
M = max length of the key

For example, when you want to crack an half of a LanManager passwords (LM) using the character set "ABCDEFGHIJKLMNOPQRSTUVWXYZ" of 26 letters, the brute-force cracker have to try KS = 26^1 + 26^2 + 26^3 + ... + 26^7 = 8353082582 different keys. If you want to crack the same password using the character set "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/", the number of keys to try rises at 6823331935124.

Exhaustive key search cracking could take a very long time to complete however if the character set is the right one the password will be cracked; its only matter of time.



How it works

The dialog offers the possibility to choose from a set of pre-defined character sets or to input a custom one; the initial password can also be changed for resuming a previous attack. The "Key Rate" field indicates the number of keys that the attack tries every second against all hashes/encrypted passwords loaded. The "Time Left" indicates the remaining time to complete the key space and the "Current password" field shows the actual key tested by the program.

The cracker's list that started the attack is updated when you exit the dialog, in order to reflect all the passwords found.

Dictionary Password Cracker

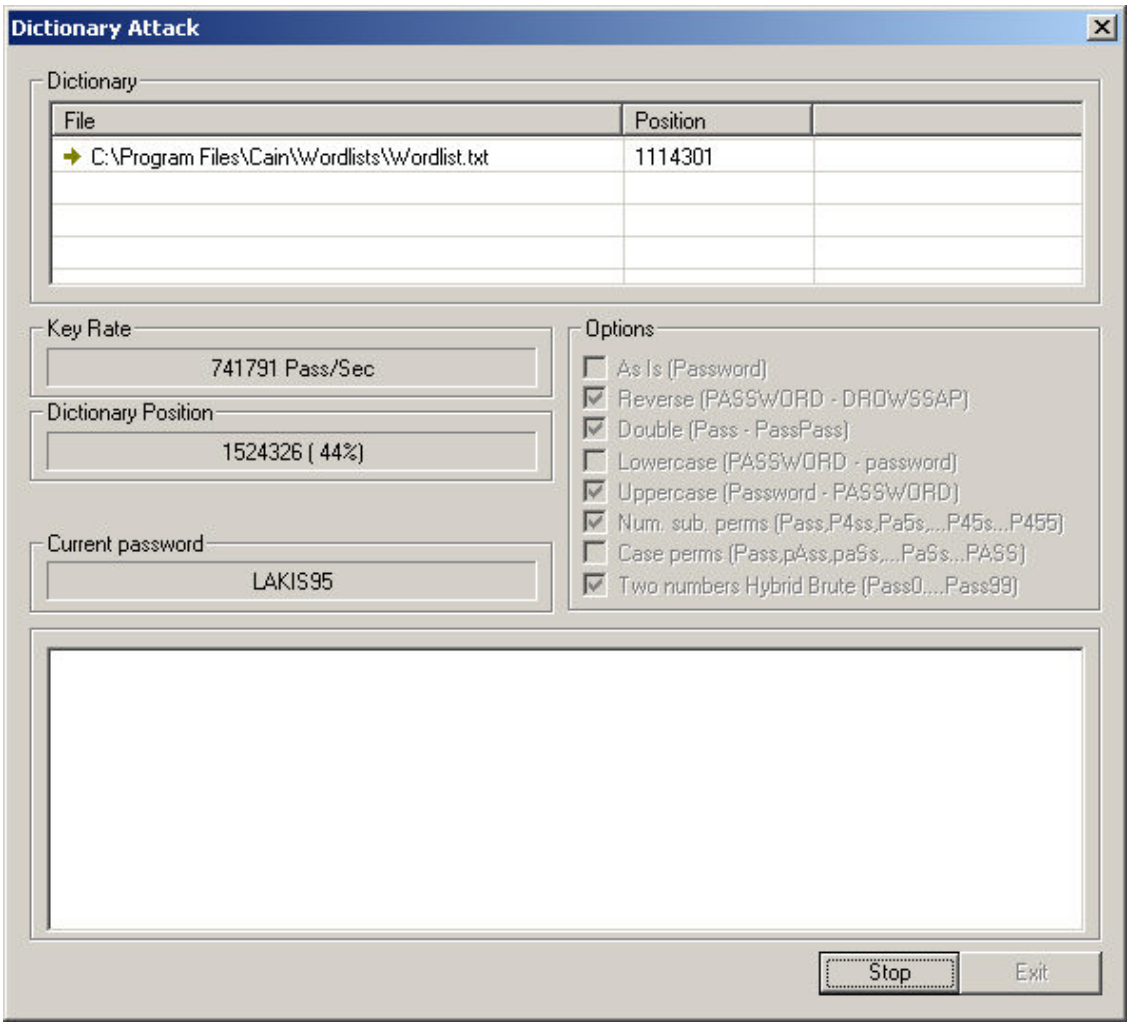
A dictionary attack consists of trying "every word in the dictionary" as a possible key for an encrypted password. A dictionary of potential passwords is more accurately known as a wordlist. This kind of attack is generally more efficient than a brute-force attack, because users typically choose poor passwords.

There are two methods of improving the success of a dictionary attack: the first method is to use a larger dictionary, or more dictionaries (technical dictionaries and foreign language dictionaries will increase the overall chance of discovering the correct password); the second method is to perform string manipulation on the dictionary.

For example, the dictionary may have the word "password" in it. Common string manipulation techniques will try the word backwards (drowssap), appending numbers to the end of the string (password00 - password99), or with different capitalization (Password, pAssword, ... passworD).

Cain's Dictionary Password Cracker can be configured to use a list of dictionary files and it also offers the possibility to apply a number of variants for each word:

- As Is: -> the password id checked as written in the dictionary file.
- Reverse -> the reverse form of the password is tried (password->drowssap).
- Double -> double the tried password (Pass->PassPass)
- Lowercase -> the lowercase form of the password is tried (Password -> password).
- Uppercase -> the uppercase password is tried (Password -> PASSWORD).
- Numbers substitution permutations -> replace certain letters with numbers (Pass, P4ss, Pa5s, ... P45s, ...P455).
- Case Perms -> all case permutation of the password are checked (password, Password, pAssword, PAssword, ... PASSWORD).
- Two numbers Hybrid-Brute: appends a maximum of two digit after each word (Password0, Password1,...Password9, Password00, Password01, ... Password99).



It also remembers each dictionary file position reached to resume from previous attacks (the Reset button cleans the start position of the wordlists).

Usage

Add all the dictionary files using the "Add" button. Variants for each word can be enabled/disabled clicking on the relative check boxes and the attack is started using the "Start" button.

The cracker's list that started the attack is updated when you exit the dialog, in order to reflect all the passwords found.

Cryptanalysis

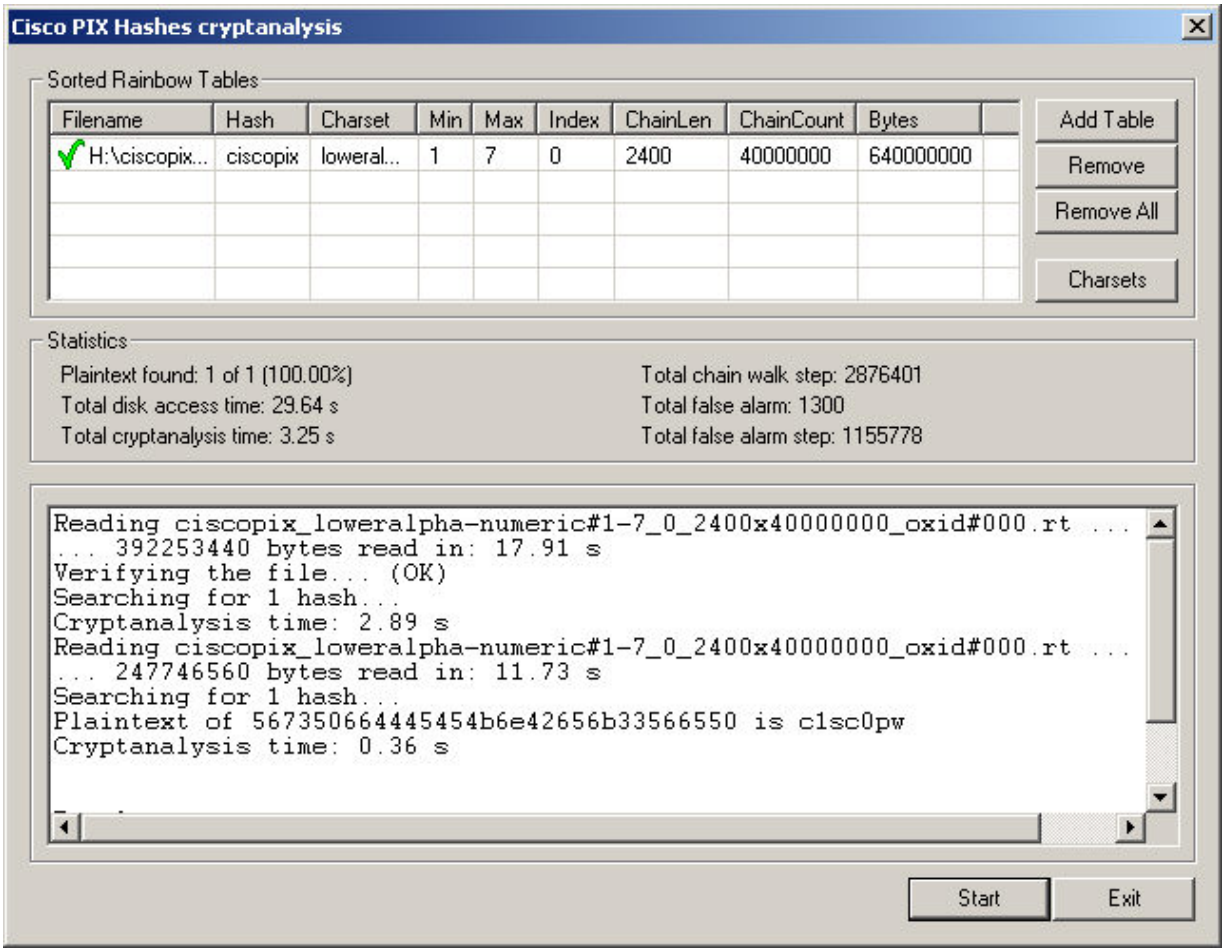
This feature enables password cracking using the 'Faster Cryptanalytic time – memory trade off' method introduced by Philippe Oechslin. This cracking technique uses a set of large tables of pre-calculated encrypted passwords, called Rainbow Tables, to improve the trade-off methods known today and to speed up the recovery of clear text passwords.

It is fully compatible with the well known software RainbowCrack by Zhu Shuanglei, the first software implementation of the above algorithm, and supports Rainbow Tables for the following hashing/encryption algorithms: LM, FastLM, NTLM, CiscoPIX, MD2, MD4, MD5, SHA-1, SHA-2 (256), SHA-2 (384), SHA-2 (512), MySQL (323), MySQL (SHA1), RIPEMD160.

Rainbow Tables can be generated using the "rtgen.exe" program, included with RainbowCrack, or the Windows "winrtgen" utility available at www.oxid.it.

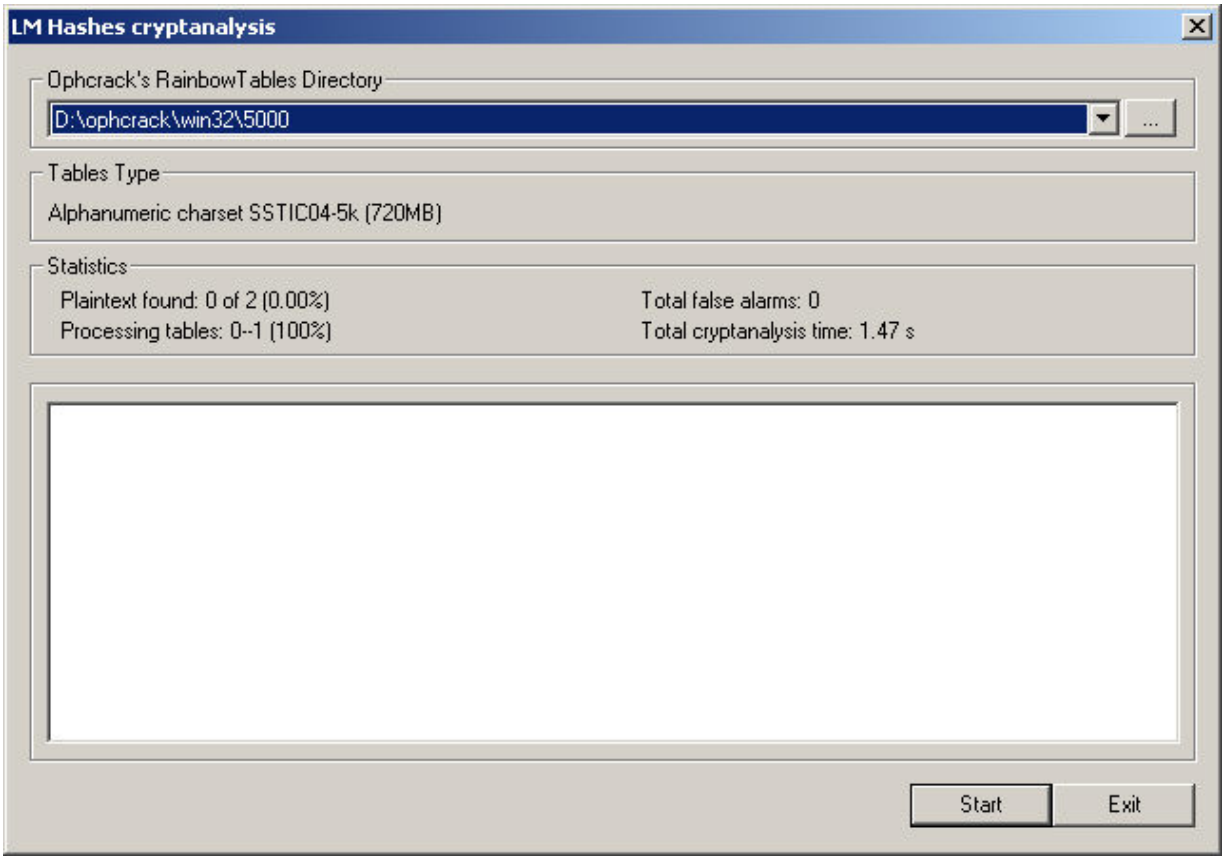
This cracking technique is pretty fast however it is useful to crack only some kind of encrypted passwords only. In challenge-response authentication protocols for example, a variable length array of bytes (the challenge) is encrypted using a key derived from the user's password. The challenge varies at each authentication so even if the user inputs the same password, two encrypted hashes are always different. The same thing happens if the encrypted password is generated using a variable "salt", which provides some sort of perturbation in the algorithm; to successfully crack a password in the above situations you should generate different Rainbow Tables for each challenge/salt used and this is really impractical.
Please note that the majority of modern network protocols already use the challenge-response mechanism so, generally speaking, this attack is not suitable for password hashes captured from the network; on the contrary it is really effective to crack straight hashes often used to store encrypted passwords locally.

The following image shows the recovery of a Cisco PIX Firewall "enable mode" password using the cryptanalysis attack:



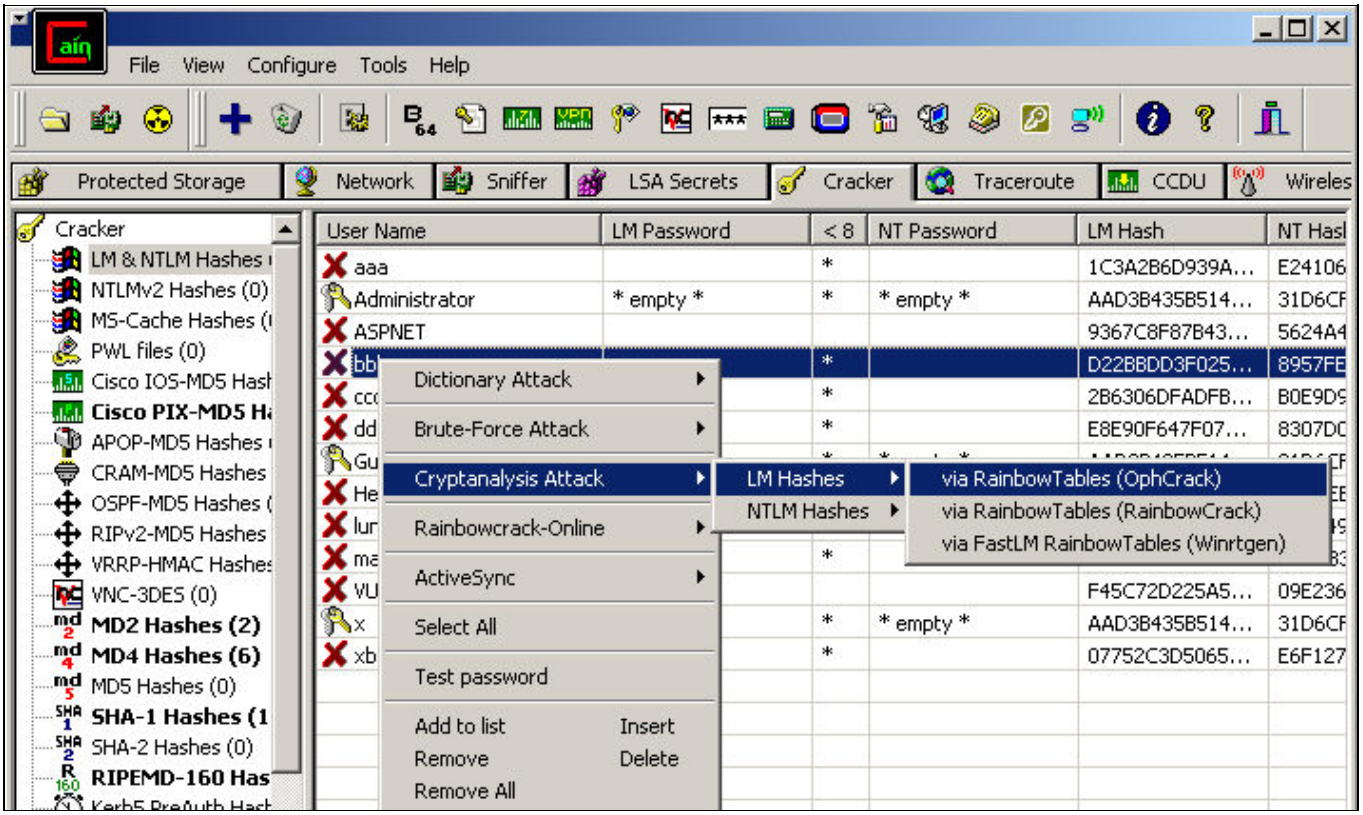
As you can see the above password (c1sc0pw), encrypted by the PIX using the MD5 algorithm, has been cracked in 3.25 seconds only.

From version 2.9 Cain is also compatible with Ophcrack's RainbowTable format (http://www.objectif-securite.ch/ophcrack/). Ophcrack's tables are more compact then those used by RainbowCracks and they are freely available at the following link: http://lasecwww.epfl.ch/~oechslin/projects/ophcrack.



Usage

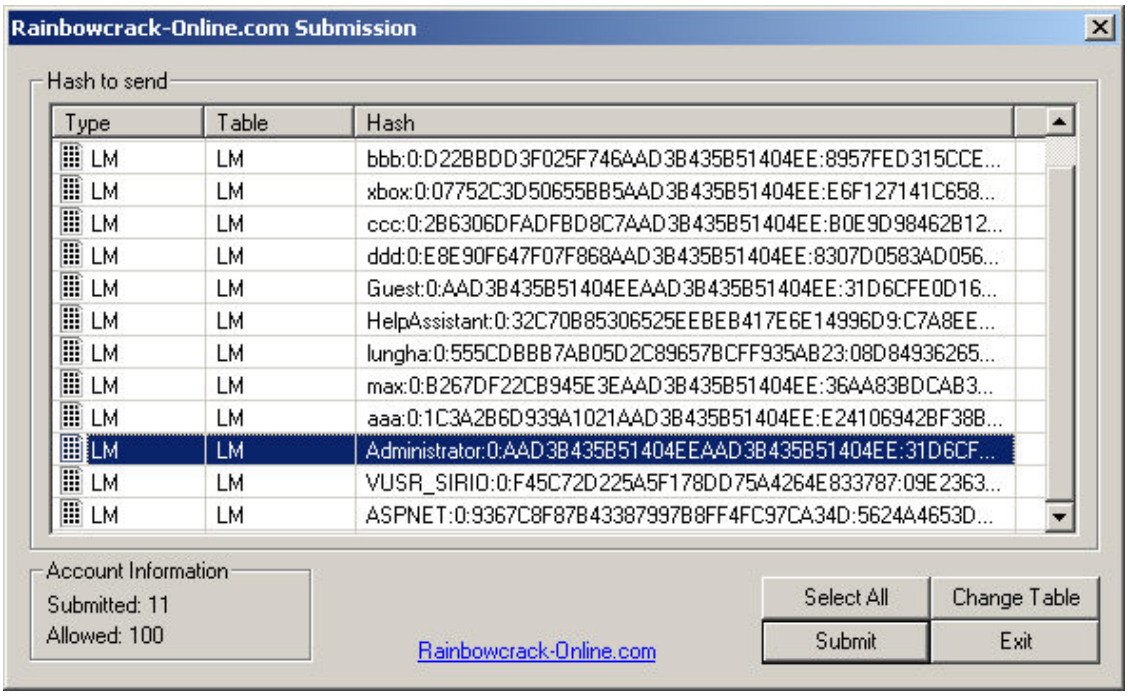
Cain's cryptanalysis attack can be launched, where possible, using the cracker's list pop up menu as illustrated below:



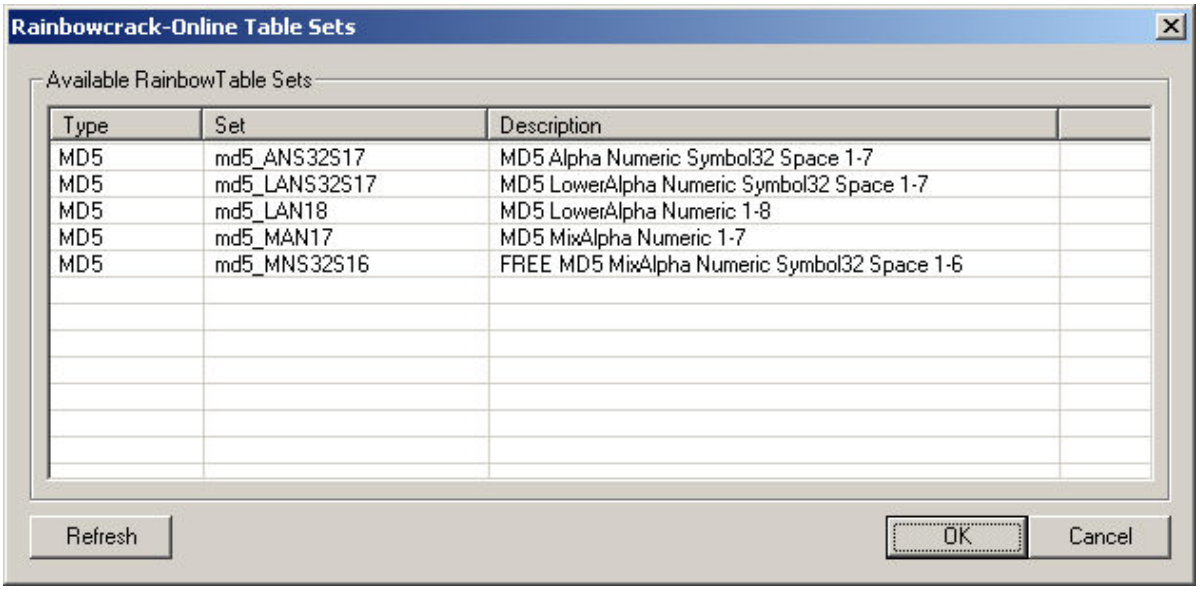
The program will load all selected hashes, into the cracking dialog, accordingly to the type of attack chosen. When using RainbowCracks tables, custom charsets can be loaded from a RainbowCrack's compatible "charsets.txt" file by mean of the relative dialog button.

Rainbowcrack-online

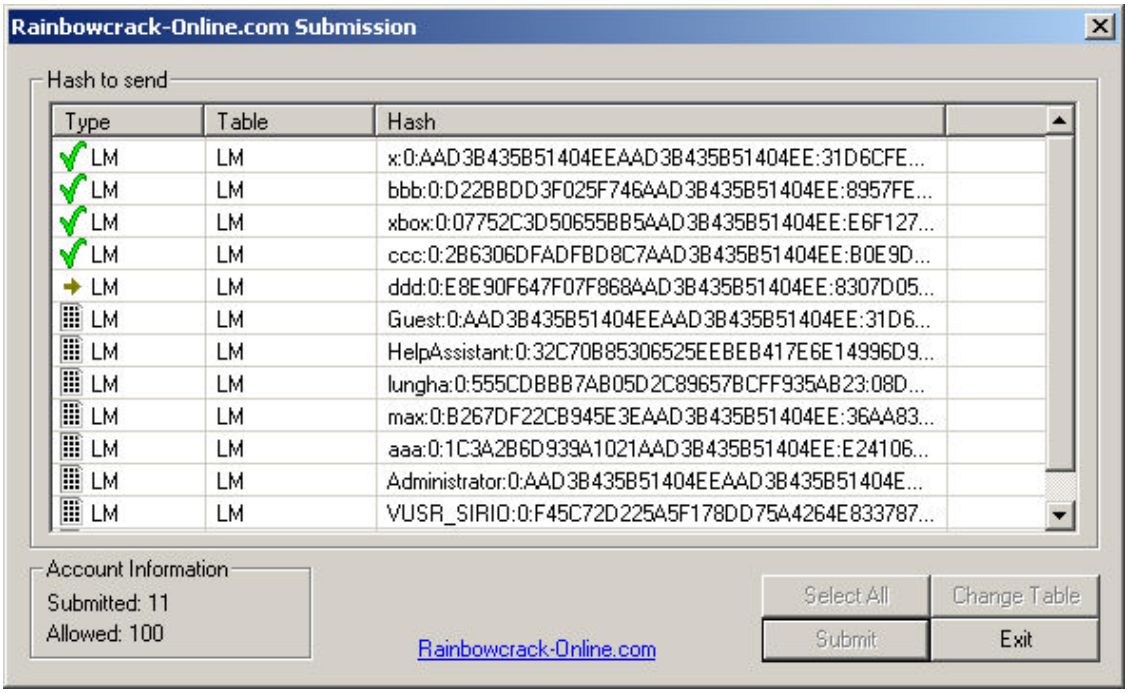
Rainbowcrack-online.com enables businesses and individuals to assess their password policies by providing access to their pre-generated hash tables.
Cain can now interact with the outstanding power of this on-line cracking service based on RainbowTable technology, but **be warned that the service is not free and you need a valid account to use this feature**.



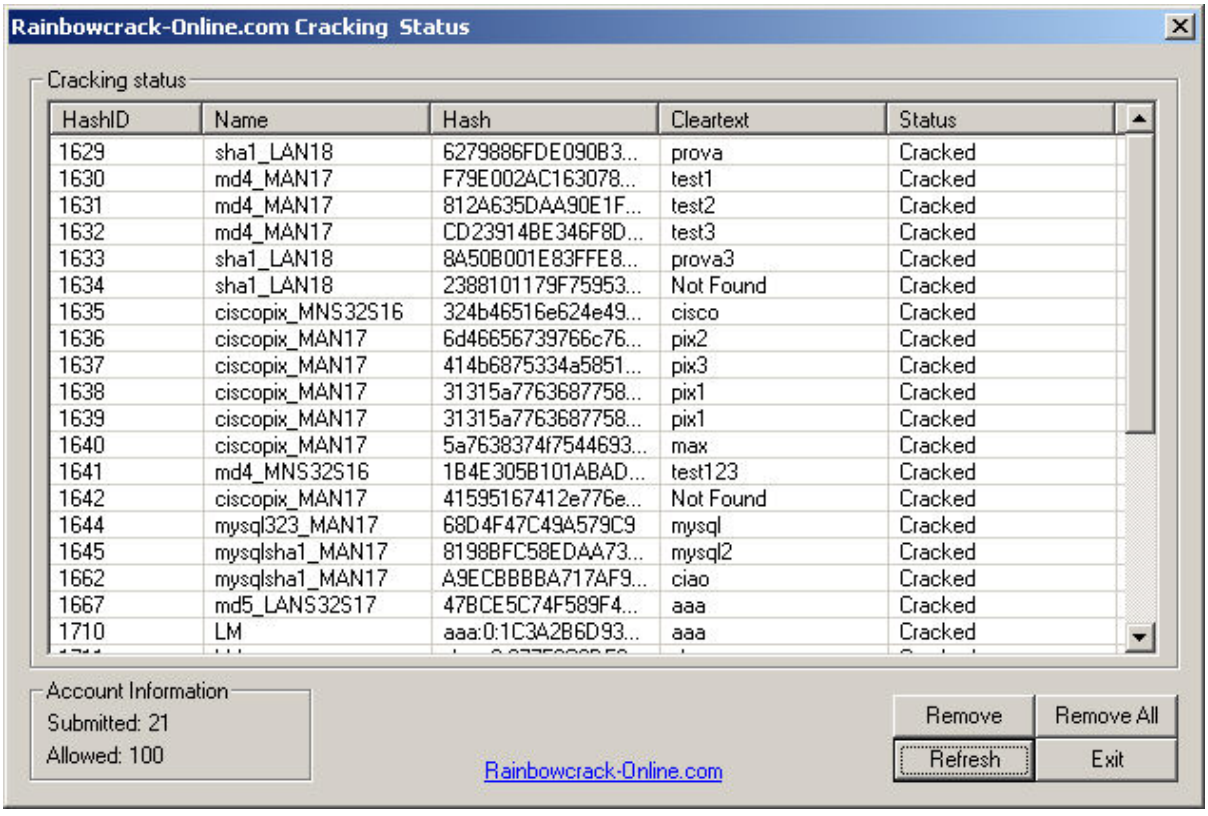
Using the Rainbowcrack-online client is really simple. You have to select one or more hashes to submit to the site and associate the correct table set for the cracking process using the "Change Table" button.



As you can see from the above picture, there could be more that one table set available for a single hash.
To submit the hashes simply select them and press the "Submit" button.



Once submitted, hashes are immediately processed by back-end systems of Rainbowcrack-online. You can check the cracking status using the "Check cracking status"
function available in Cain's hash lists.



When you exit this dialog, cleartext passwords are automatically imported into Cain's list.

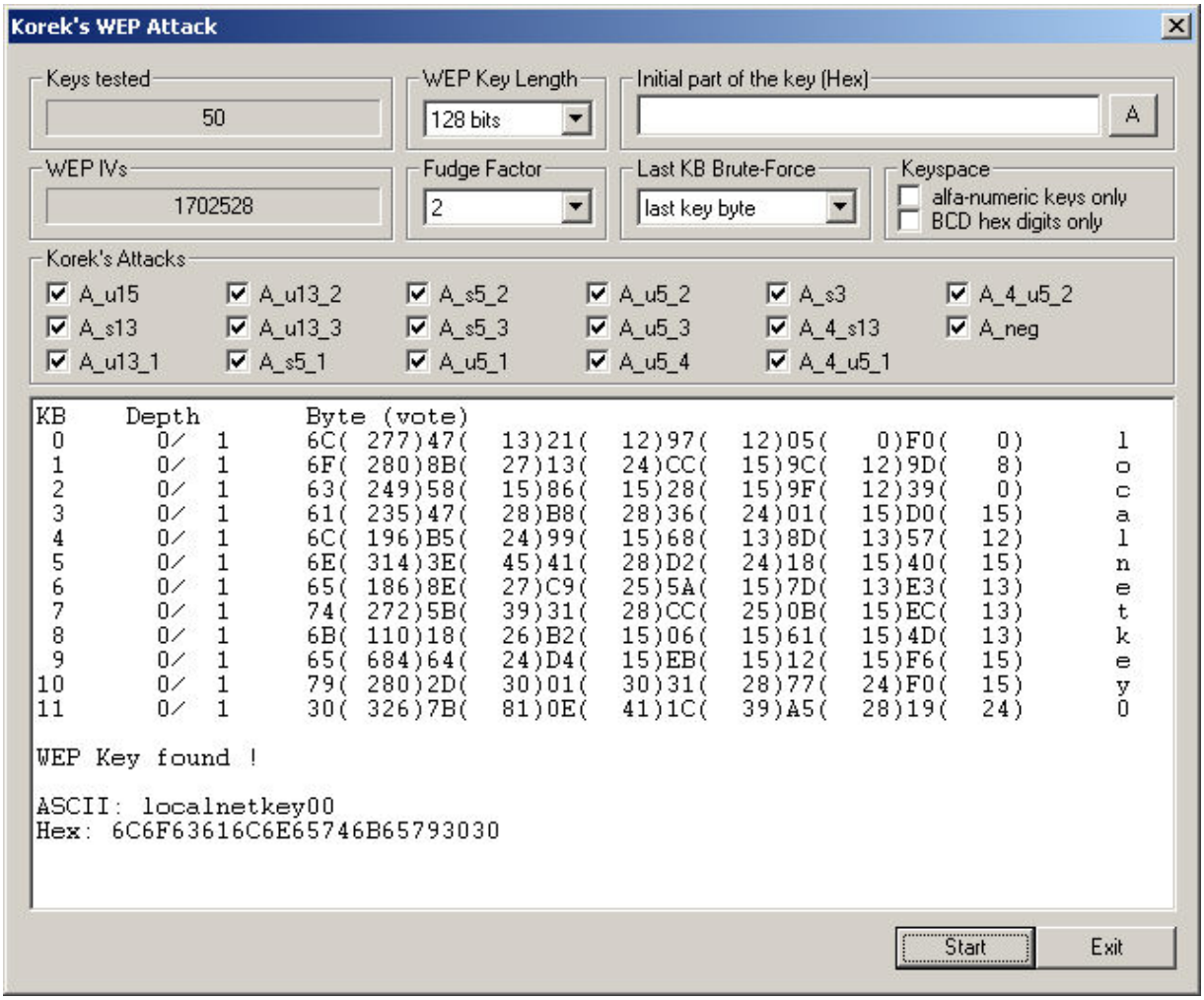
Note

The communication between Cain and the Rainbowcrack-online web site is SSL enabled to ensure privacy of transmitted information.

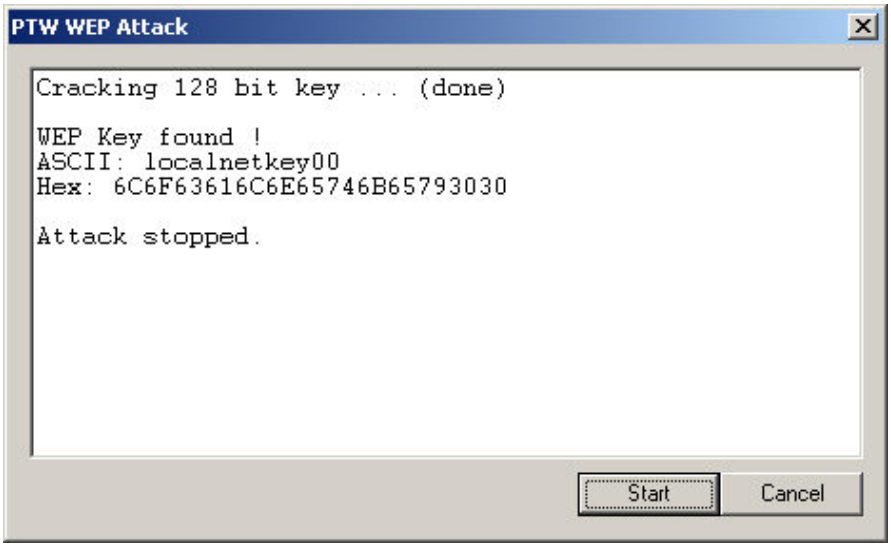
WEP Cracker

The Korek's WEP Attack is a statistical cracking method for the recovery of a WEP Key. The attack is based on some weakness of the RC4 encryption algorithm well documented in the paper "Weaknesses in the Key Scheduling Algorithm of RC4" from Scott Fluhrer, Itsik Mantin and Adi Shamir.

This feature covers the same functionality of the software Aircrack from Christophe Devine and can quickly recover 64-bit and 128-bit WEP keys if enough unique WEP IVs are available. Accordingly to Aircrack's documentation the minimum number of unique WEP IVs needed to successfully crack a WEP Key using the Korek's Attack is: 250.000 for 64-bit WEP keys and 1.000.000 or more for 128-bit keys.



Can also support the newly discovered PTW cracking method which is able to extend Klein's attack and optimize it for usage against WEP. Using this method it is possible to recover a 104 bit WEP key with probability 50% using just 40,000 captured packets. For 60,000 available data packets, the success probability is about 80% and for 85,000 data packets about 95%. Using active techniques like deauth and ARP re-injection, 40,000 packets can be captured in less than one minute under good condition.

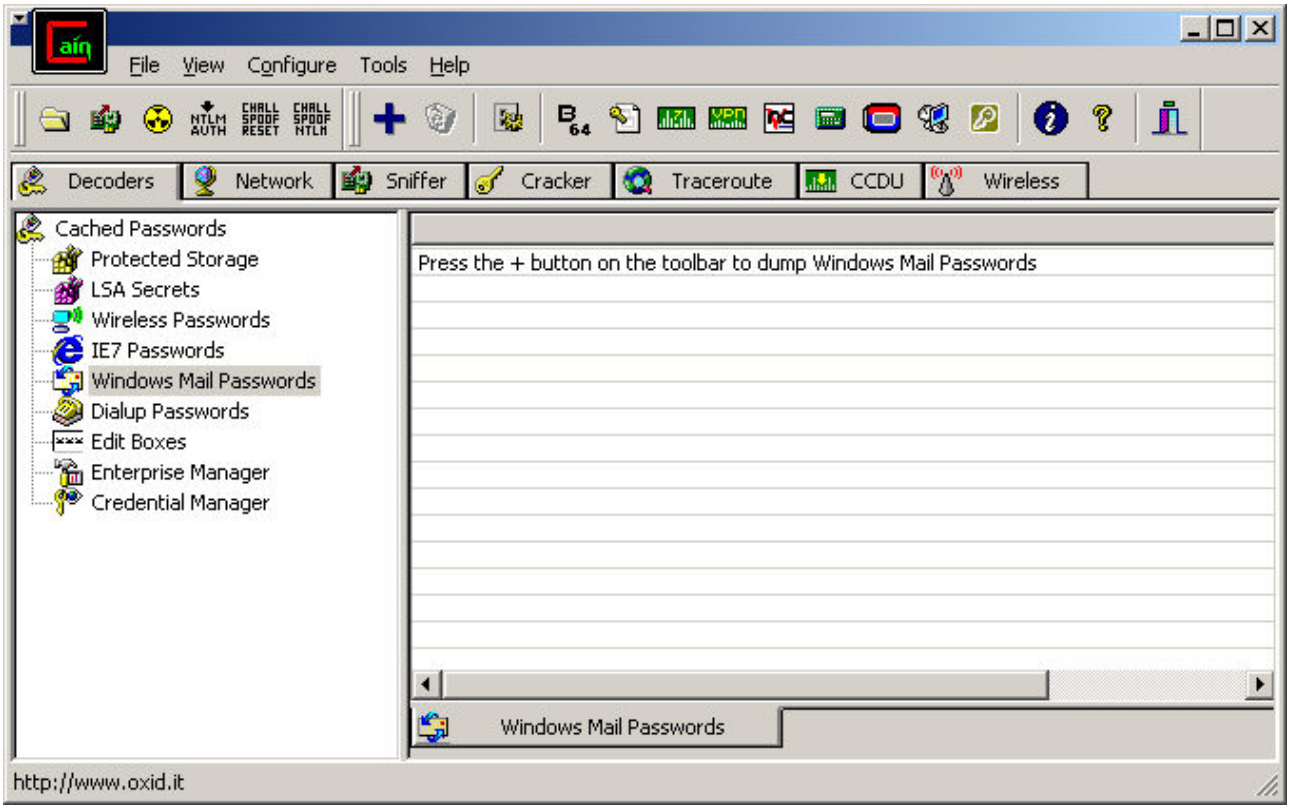


Requirements:

The attack can fails for different reasons:
- more WEP IVs needed (you have to capture more WEP IVs).
- the network is using a dynamic WEP key.
- the capture file is corrupted (equal or negative votes).
- false positives (try to disable some Korek's attack unchecking the relative checkbox and raise the attack's "Fudge Factor").
- wrong key length set (there is no way to know the key length from wireless network packets so try the attack twice, first set a 64-bit key and then longer).

Password Decoders

Password Decoders can be used to immediately decode encrypted passwords from several sources, for example the Windows Protected Store, the Credential Manager, standard Edit Boxes, LSA secrets, passwords from SQL Enterprise Manager, Windows Mail, DialUp, Remote Desktop profiles and the Windows wireless configuration service.
Cain also includes decoders for non Microsoft applications like VNC, Cisco VPN client profiles ..... every encrypted information that can be recovered instantly is managed by the program with the appropriate “decoder”.



These features usually require local access on the target system, anyway Cain also provide the capability to extract some information from external Windows registry hive files.

Access Database Passwords Decoder

This feature reveals the password used to protect database files (.MDB) created with Microsoft Access 95/97/2000/XP.

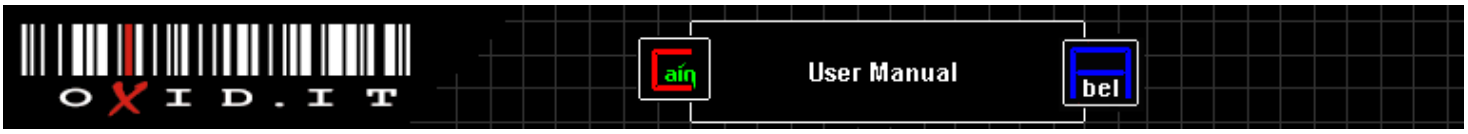
The main password is stored in database files using a simple XOR encryption that can be reversed immediately.

Limitations

This tool recovers the main database password only; user level passwords are not shown.

Usage

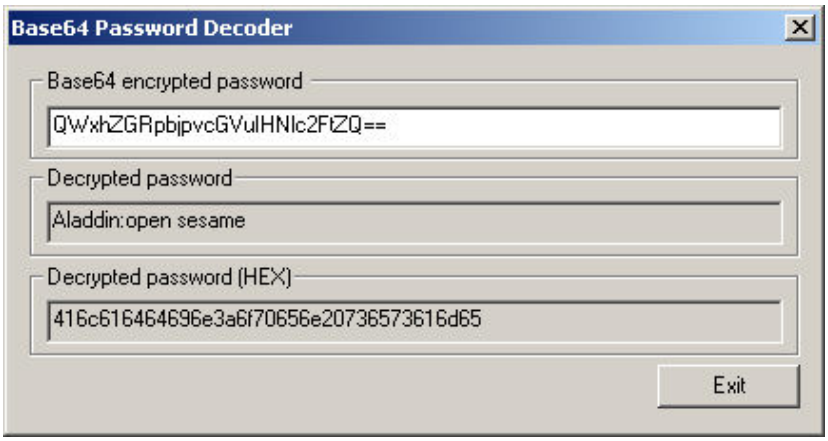
Everything is automatic, you have only to select the password protected database.



Base64 Password Decoder

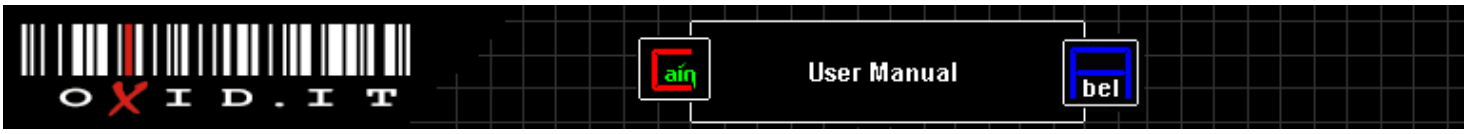
This feature decodes Base64 encrypted strings to their original byte form. Base64 is used in several Internet protocol standards like HTTP, MIME, IMAP .... to encode arbitrary data as plain ASCII text. The following picture shows the decoding of a Base64 password captured from an HTTP Basic authentication packet:

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==



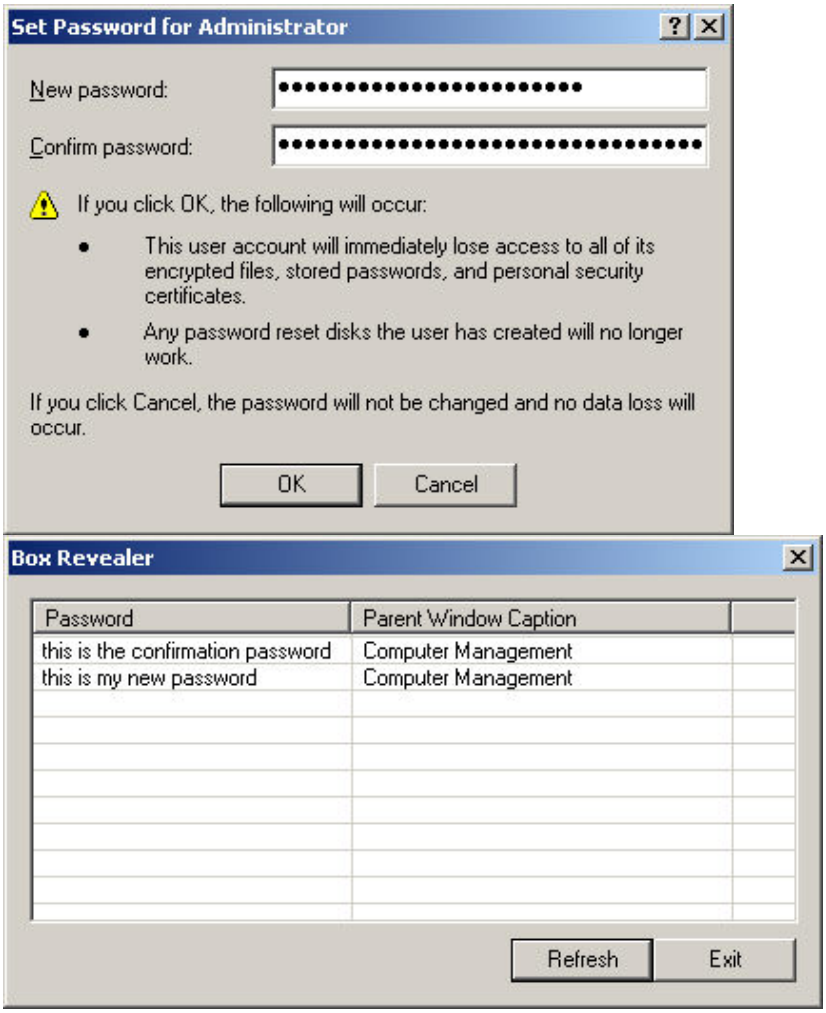
Usage

Base64 Password Decoder dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button. Simply cut and paste the Base64 encrypted string into the dialog, the decoder will do the rest.



Box Revealer

This feature can reveal the passwords stored behind the asterisks in standard password text-boxes.



How it works

This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to dump passwords hidden behind asterisks in password text-boxes. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed the thread will run with the same access privileges of the Local Security Authority Subsystem; it loads the function "DumpBox" from Abel.dll which enumerates every password text-box present on the screen and dumps its text (the password) into a temporary file named boxes.txt. Finally, the content of this file is put on the screen and the temporary file is deleted.

The "Box Revealer" supports most standard password text-boxes, however some applications don't store passwords behind the asterisks for security reasons. In such cases this feature will not be able to show the passwords.

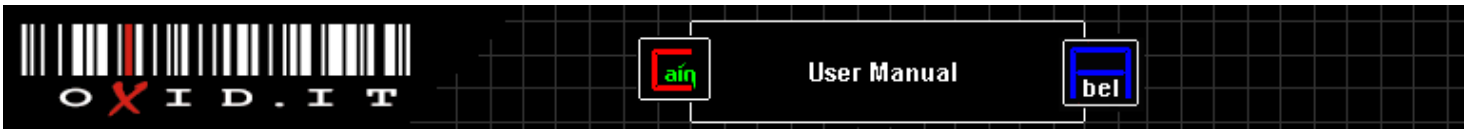
Usage

To dump passwords hidden behind asterisks you can press the "Insert" button on the keyboard or click the icon with the blue + on

the toolbar.

Requirements

This feature requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Abel.dll is also required by the remote thread injected into LSASS process.



Cisco Type-7 Password Decoder

Cisco devices can use a proprietary encryption algorithm to encrypt the password for enable mode and vty lines. This kind of encryption is used when "**service password-encryption**" has been enabled on the device and produces, what they call, Type-7 passwords.

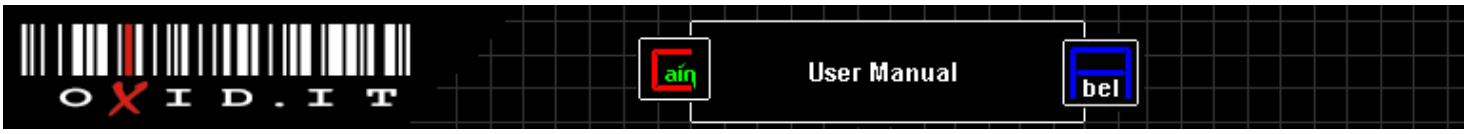
This feature allows you to decode encrypted passwords from configuration files.

Limitations

This tools cannot decode Cisco Type-5 passwords. You can use Cain's IOS-MD5 Cracker to do that.

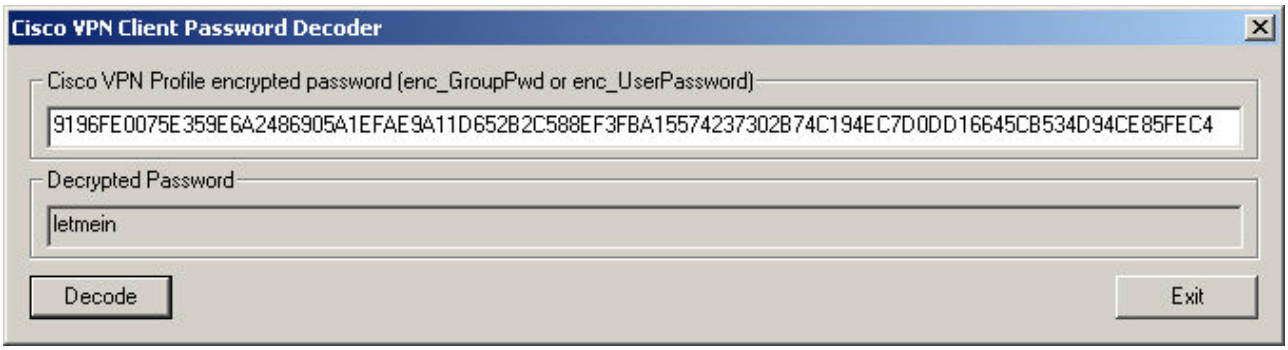
Usage

Simply cut and paste the encrypted password into the dialog; the decoder will do the rest.



Cisco VPN Client Password Decoder

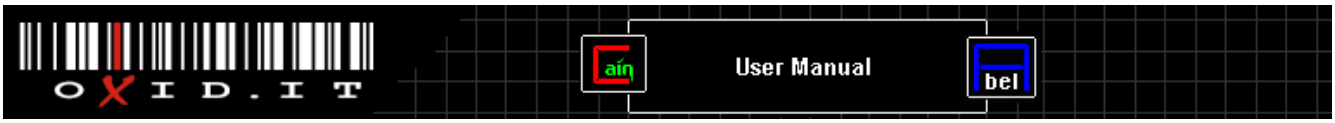
This feature decrypts passwords stored in profile files (*.pcf) by Cisco VPN Client software. These files contain all the parameters needed to connect to a remote network via VPN tunnels.



Keys generated by Cisco software's encryption algorithm are created using SHA1 and 3DES in EDE mode. They are user and machine independent.

Usage

Simply cut and paste the encrypted password (enc_GroupPwd or enc_UserPassword) into the dialog; the decoder will do the rest.



Credential Manager Password Decoder

Introduction

Credential Manager is a new SSO solution that Microsoft offers in Windows Server 2003 and Windows XP to provide a secured store for credential information. It allows you to input user names and passwords for various network resources and applications once, and then have the system automatically supply that information for subsequent visits to those resources without your intervention.

For example when you use the command:
*net use * \\computer_name\share_name /user:user_name password /savecred*

Credential Manager stores the supplied password in the so called "Enterprise Credential Set" of the local machine.

This set of credentials is stored in the file
\Documents and Settings\%Username%\Application Data\Microsoft\Credentials\%UserSID%\Credentials
and is encrypted using the DPAPI subsystem.

There is also another set used for credentials that should persist on the local machine only and cannot be used in roaming profiles, this is called "Local Credential Set" and it refers to the file:
\Documents and Settings\%Username%\Local Settings\Application Data\Microsoft\Credentials\%UserSID%\Credentials

You can find more information about this feature at the following link:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/dpapiusercredentials.asp

Microsoft Windows XP/2003 exports some APIs that application developers can use to interact with the Credential Manager; for example the function

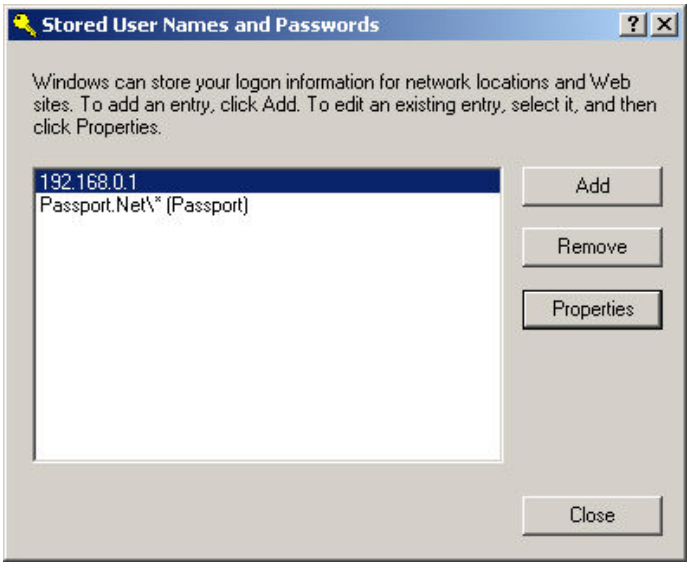
BOOL CredEnumerate (LPCSTR Filter, DWORD Flags, DWORD Count, PCREDENTIAL** Credentials)*

can be used to enumerate all cached resources and passwords but only some of these credentials can be viewed in clear text.
Accordingly with the MSDN documentation, Credential Manager can store different types of credentials under the form of passwords, security BLOBS or certificate files.

Value	Description
CRED_TYPE_GENERIC	The credential is a generic credential. The credential will not be used by any particular authentication package. The credential will be stored securely but has no other significant characteristics.
CRED_TYPE_DOMAIN_PASSWORD	The credential is a password credential and is specific to Microsoft's authentication packages. The NTLM, Kerberos, and Negotiate authentication packages will automatically use this credential when connecting to the named target.
CRED_TYPE_DOMAIN_CERTIFICATE	The credential is a certificate credential and is specific to Microsoft's authentication packages. The Kerberos, Negotiate, and Schannel authentication packages automatically use this credential when connecting to the named target.
CRED_TYPE_DOMAIN_VISIBLE_PASSWORD	The credential is a password credential and is specific to Microsoft's authentication packages. The Passport authentication package will automatically use this credential when connecting to the named target. Additional values will be defined in the future. Applications should be written to allow for credential types they do not understand.

If you use the above function to enumerate a credential of type CRED_TYPE_DOMAIN_PASSWORD for example, the API will not return the cached password in clear text form.

Non-developer users can interact with Credential Manager using the application "Stored User Names and Passwords" that can be found under:
Start-> Settings-> Control Panel-> User Accounts-> %Account% -> Manage my network passwords.



As you can see from the image above this application cannot show you what the passwords cached into your system are.

How it works

This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to decrypt credential files. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed, the thread will run with the same access privileges as the Local Security Authority Subsystem and will use the "DumpCF" function of Abel.dll to call the native undocumented LsaICryptUnprotectData API from LSASRV.DLL end decrypt the credential's files. The thread stores the output of this API in a temporary file named cred.txt located in the same directory of the program. Finally, user's credentials are dumped and put on the screen.

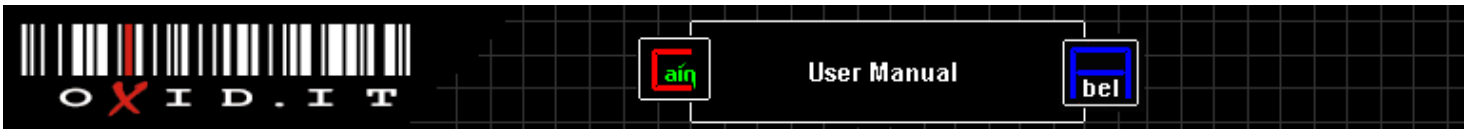
Credential Manager can store various kinds of passwords; they can be saved as MultiByte or WideChar strings, security BLOBS and certificates too. The choice of the final encryption method is left to the user.
The program will try to recognize plain text passwords stored as MultiByte strings or WideChar strings, and will also decode Passport and Standard (no entropy) credential BLOBS originally encrypted using the CryptProtectData API.

Usage

Credential Manager Password Decoder dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button.

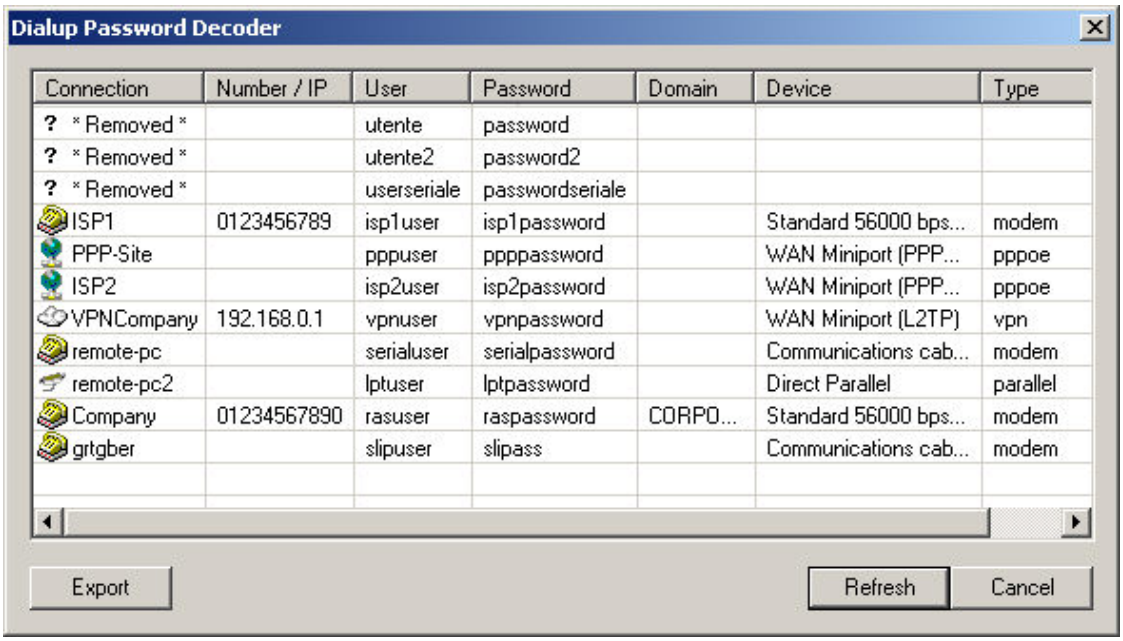
Requirements

This feature requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Abel.dll is also required by the remote thread injected into LSASS process.



Dialup Password Decoder

The Dial-Up Password Decoder reveals passwords stored by Windows "Dial-Up Networking" component. RAS credentials are usually stored in LSA Secrets "**L$_RasDefaultCredentials**" and "**RasDialParams!<UserSID>**" while all other connection parameters (phone number, ip address....) reside into Phonebook files (.pbk).



The information contained in the list can also be exported into text files by pressing the "Export" button.

How it works

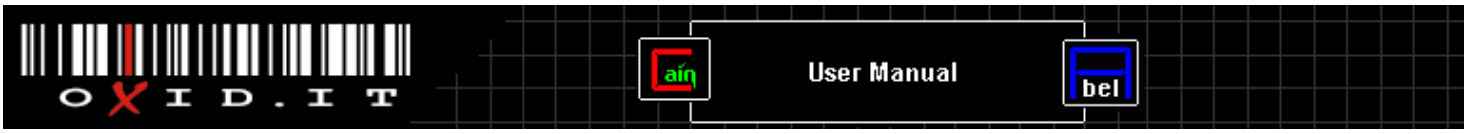
This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to dump LSA secrets present on the system. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed the thread will run with the same access privileges of the Local Security Authority Subsystem; it will load the function "DumpLsa" from Abel.dll which will open and query each secret using the LsarOpenSecret and LsarQuerySecret APIs from LSASRV.DLL. The thread stores the data returned from these functions in a temporary file named lsa.txt located in the same directory of the program. Finally the program extracts from the temporary file all the credentials related to "Dial-Up Networking"
associating them with the parameters found in Phonebook files.

Usage

Dial-Up Password Decoder dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button.

Requirements

This feature requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Abel.dll is also required by the remote thread injected into LSASS process.



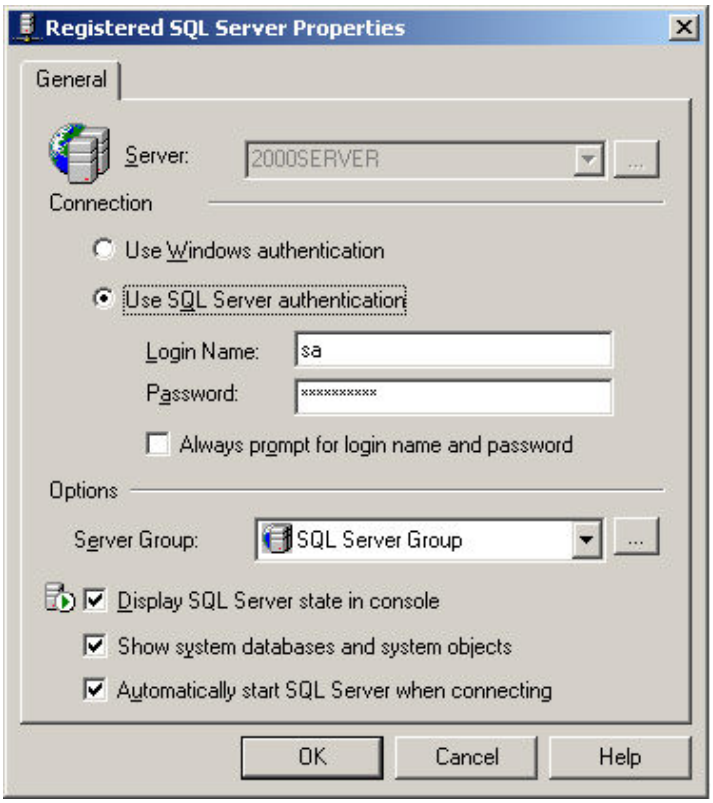
Enterprise Manager Password Decoder

Enterprise Manager is a Windows application used to manage Microsoft SQL Server 7.0 and 2000. When configured to use SQL Server authentication, Enterprise Manager stores the connection credentials in the registry under the key

SQL2000: *HKEY_CURRENT_USER\\Software\\Microsoft\\Microsoft SQL Server\\80\\Tools\\SQLEW\\Registered Servers X*

and

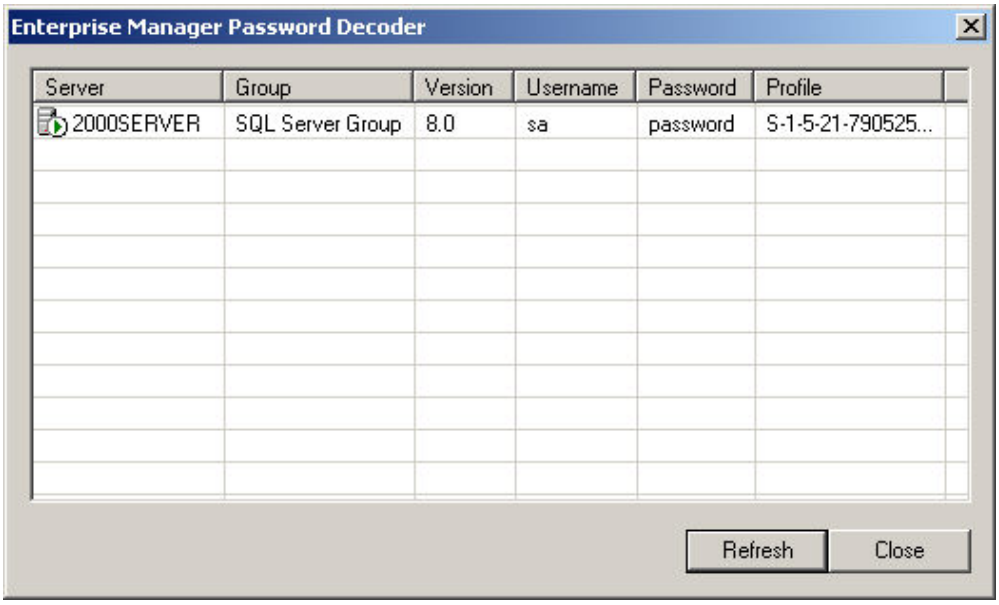
SQL 7.0: *HKEY_CURRENT_USER\\Software\\Microsoft\\MSSQLServer\\SQLEW\\Registered Servers X*



SQL credentials are encrypted/decrypted using a simple XOR encryption algorithm illustrated below:

```
void xor_cred (char *credential, int len)
{
	BYTE XorTable[]={0x67,0x66,0x3b,0x64,0x6b,0x6c,0x67,0x5e,0x25,0x24,0x5e,0x6c,0x3b,0x64,0x73,0x27,0x63,0x00};
	int xorlen = strlen ((char*) XorTable);
	for (int j=0,i=0; i<len; i++,j++)
	{
		if (j==xorlen) j=0;
		credential[i] ^= XorTable[j];
	}
}
```

SQL Server 2000 further protects the XOR encrypted credentials, before writing them into the registry, using the **CryptProtectData** function of "Crypt32.dll". By mean of this API, credentials can be decrypted only by the same user that previously created them and on the same machine too.



This feature decodes Enterprise Manager passwords from SQL Server 7.0 and 2000 stored in the registry.

How it works

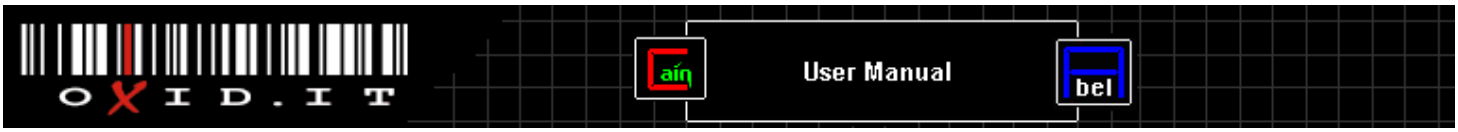
If needed, it uses the "**CryptUnprotectData**" API from CRYPT32.DLL to decode the password; this function is called without entropy. At this point it performs the XOR decryption using the above algorithm.

Usage

Cain's Enterprise Manager Password Decoder dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button.

Requirements

This tool requires to be executed on the same machine where the password was created and with the same user account too.



Protected Storage Password Manager

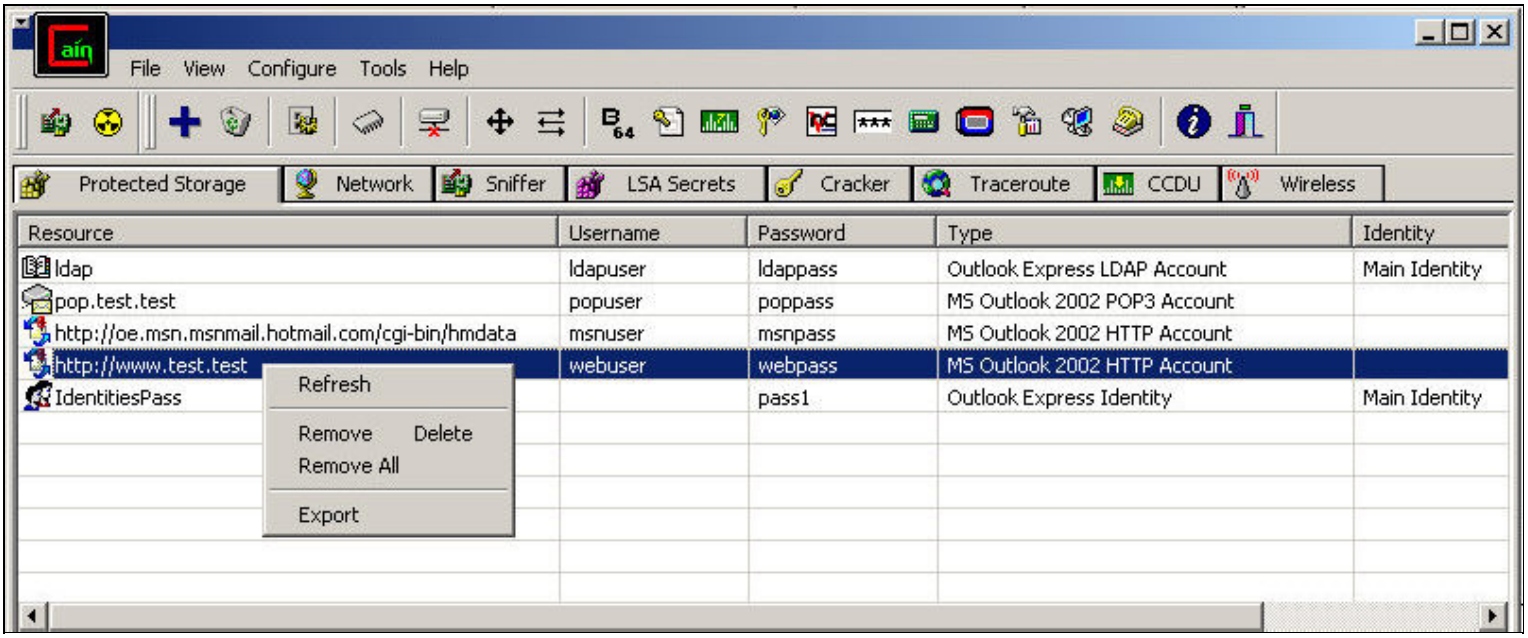
The Protected Store is a storage facility provided as part of Microsoft CryptoAPI. It's primarily use is to securely store private keys that have been issued to a user. All of the information in the Protected Store is encrypted, using a key that is derived from the user's logon password. Access to the information is tightly regulated so that only the owner of the material can access it.

Many Windows applications use this feature; Internet Explorer, Outlook and Outlook Express for example store user names and passwords using this service.

Credentials are stored in the registry under the key
HKEY_CURRENT_USER\Software\Microsoft\Protected Storage System Provider
which is the base key for the service.

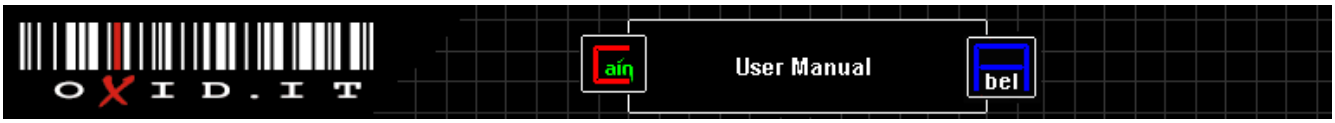
This feature enumerates those entries and decodes the following type of credentials:

- MS Outlook 2002's passwords(POP3, SMTP, IMAP, HTTP)
- Outlook Express's passwords(POP3, NNTP, SMTP, IMAP, HTTP, LDAP, HTTP-Mail)
- Outlook Express Identities
- MS Outlook's passwords (POP3, NNTP, SMTP, IMAP, LDAP, HTTP-Mail)
- MSN Explorer's Sign In passwords
- MSN Explorer's Auto Complete passwords
- Internet Explorer's protected sites passwords
- Internet Explorer's Auto complete passwords



Usage

To dump the content of the store you can press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. The Protected Storage Password Manager also lets you remove unwanted/forgotten cached passwords in a simple way: just right click on the resource in the list and choose remove from the pop up menu.

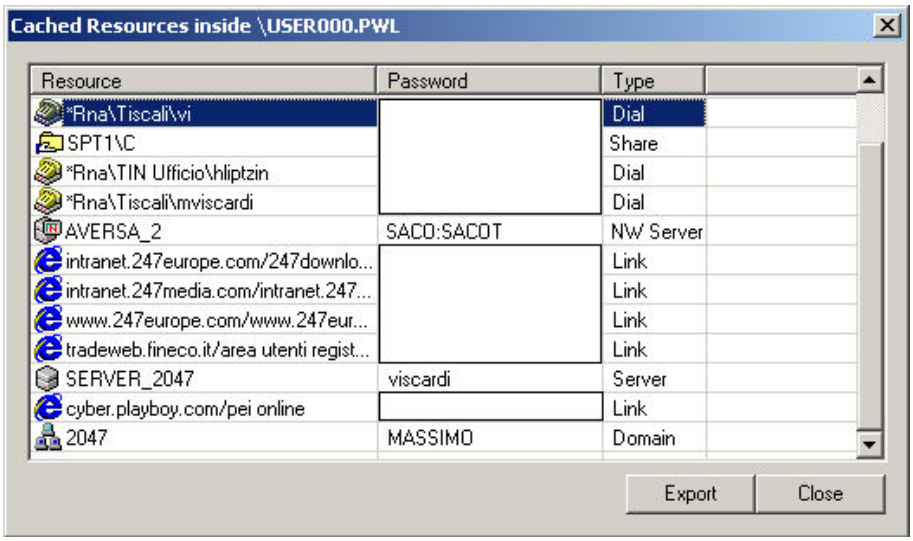


PWL Cached Password Decoder

Keeping track of multiple passwords can be a problem for users. Often, they either forget the passwords or write them down and keep lists of passwords near their computers. Windows 95/98 solves this problem by storing passwords for resources in a password list file (.PWL). If password caching is enabled (default) and a user types
and saves a password when connecting to a password-protected resource, Windows 95/98 will cache the password in the password list file.

Windows 95/98 comes with an administration utility called Password List Editor (PWLEDIT). It allows you to view the resources listed in a user's password list file but It does not allow you to view the actual passwords. If problems are encountered using a cached password you can only remove specific password entries without knowing if the password was the real problem. PWLEDIT works only if the password list file is unlocked, that is, if the user is logged on. It can be used to view the contents of the logged-on user's password list file only, so it should be executed on the user's computer.

On the contrary, Cain allows you to view all cached resources and relative passwords in clear text either from locked or unlocked password list files. That means all the passwords stored in external .PWL files can be viewed even if you are not logged-on with that file.

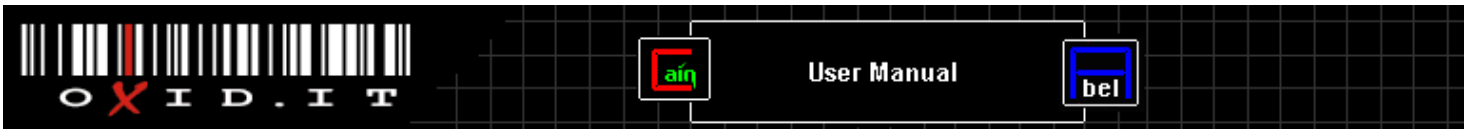


Note

In order to recover cached resources in .PWL files you must know the main user name and password for that file. Cain's PWL Password Cracker will help you to find the main password.

Usage

Once the main PWL password is found you can activate the decoder using the relative function from the list pop up menu.

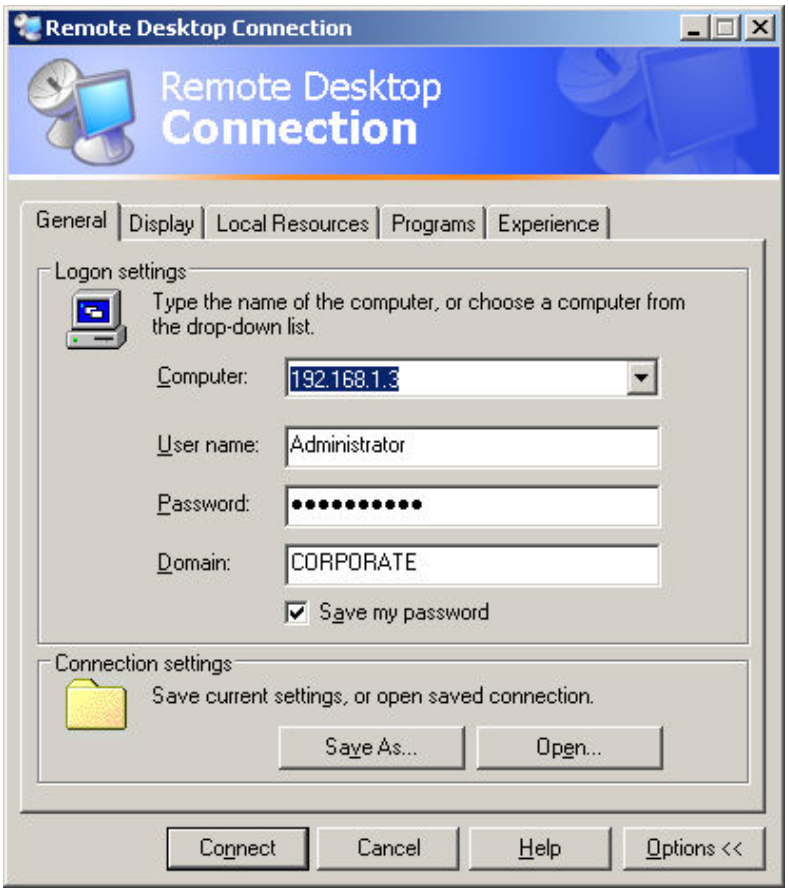


Remote Desktop Password Decoder

Remote Desktop allows you to save connection parameters in files with .RDP extensions. When you check "Save my password" in the connection dialog, the supplied credentials are encrypted using the **CryptProtectData** API and saved as well. An example of encrypted password from a test Remote Desktop Profile follows:

password 51:b:01000000D08C9DDF0115D1118C7A00C04FC297EB01000000F471161752360D4496EDA14F658FDF8E000000
0008000000700073007700000003660000A8000000100000000188EFBFDCB0C5EC2CE831603376EAC390000000004800000A
0000000100000000C568F97E208D0F4431972967599F8CB08020000EA394AFF35DB15237E0565D4D152553FB1B56B166C4F
D4D4139D41E06864924F8CF6D62A8D5A022BD57FA6CC9FE596D7FF44C182A7BE289FA8F7104F974919F4AD821C717571A6
AC962B020460A02BE6B18912B24FC356C0B3E7B966D50DE6D80D28C80B1E7EE449758D592C9B90D1A9E4389625F6D0BD0
C77CF0E78BF25C6B9B8664C2CF71BFF597A5CC4F695FD2D969C241E18976F3BC0857728A9CAEAE5DD0D5046F7173274E0
CD071945A0428BA5EA36908B0928EED7B496DD6517A70531C1638A376C36E31DF682601BC4601E4D823F6E544933E3AD4B
7663CBC81443C214A16330FEE87BAB49902776AC2A2D9B7C222EE614C7D147AC8932EDABEB85341EFCE4579AE3D077F84
17E265C4CB3B26928ED6583EC76E456CDB16D768A22629DA4147769658698285C251F222F2AB180C376B2837CA83FBC825
6B5DB03A41CBC2599C0769578F9C02550743E9F3962E696FFBFD9BB22EDAFA4CAB3D072955EEB5589C2AA992D8C8FCFB2
C3665A8B30CF9CD0109FF232882B9493054565C4E26EB4566EAA7183E4CBBAB7C2A30575C3CC84946B121F0D044F6CAD20
DDFBF6647135D08CAA2D0A25281E16EE3BE19A017BEB64A6F79296CF7CEFB140C6E4BE71E66691FECBD9CBCD83EC3A748
791DE3D3E4433034F8968C71A793670E8F4CB0AC479AB5F045F514BB109ED5FCB2B1CFA8BCEB05D7E0186CD871DDB2C13
41E3E9688358D2932835B514000000E16BE7FD121813D382B2A02360FF427D0F8B23D20

Because of the usage of CryptProtectData API the above encrypted password can be decrypted only by the user that created that .RDP file and on the same machine too.



This feature decodes the password used in Remote Desktop Profiles (.RDP) generated by the Windows tool "Remote Desktop".

How it works

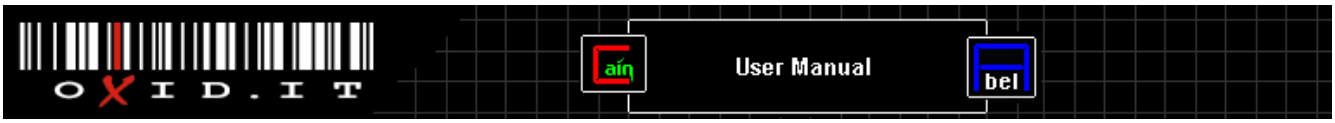
It uses the "**CryptUnprotectData**" API from CRYPT32.DLL to decode the password. The function is called without entropy.

Usage

Everything is automatic, you have only to select the Remote Desktop Profile (.RDP).

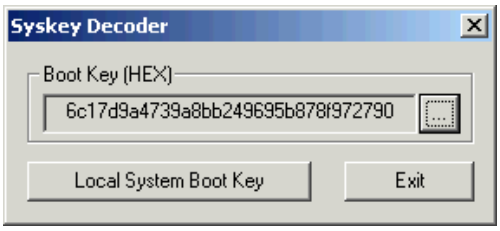
Requirements

This tool requires to be executed on the same machine where the profile was created and with the same user account too.

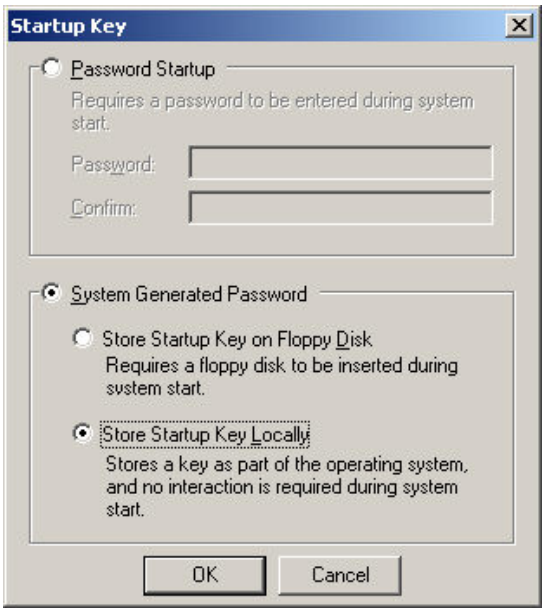


Syskey Decoder

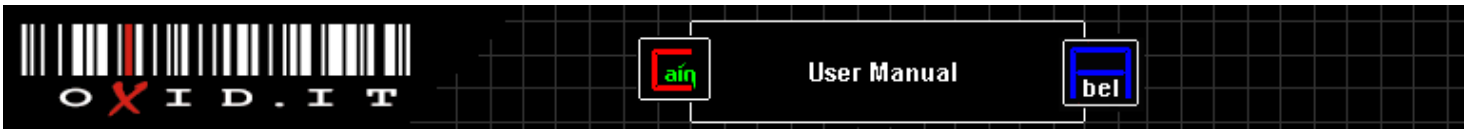
The Syskey Decoder extracts the Boot Key (Startup Key), generated by the SYSKEY utility, from the local registry or "off-line" SYSTEM files.



The Boot Key is the information used by the program SYSKEY.EXE to encrypt password hashes before they are saved to SAM database files.



If stored locally, the Boot Key is scrambled into subkeys of the following registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa.
The Syskey Decoder can reconstruct this information into its hexadecimal form, ready to be used by Cain's NT Hashes Dumper.

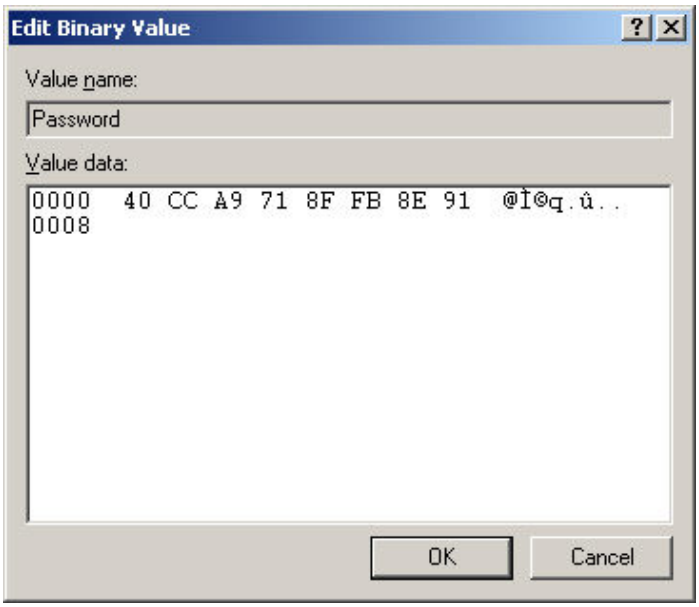


VNC Password Decoder

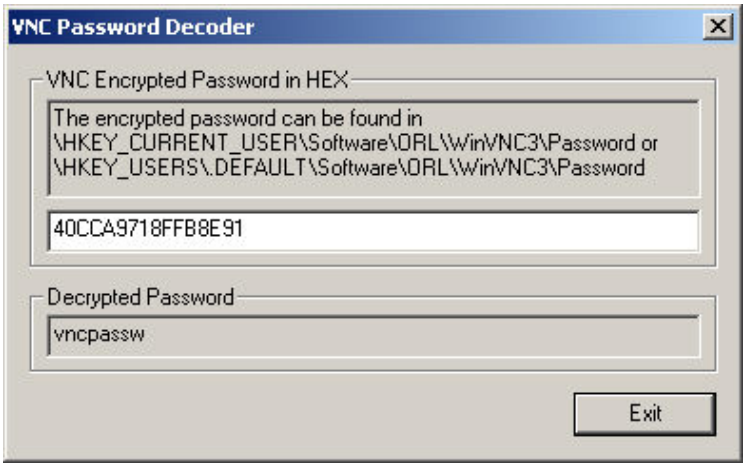
VNC (Virtual Network Computing) software is a remote desktop solution which makes it possible to view and fully-interact with one computer from any other computer or mobile device anywhere on the Internet. VNC software is cross-platform, allowing remote control between different types of computer.

The server component stores the encrypted login password in the registry under the key:

\\HKEY_CURRENT_USER\\Software\\ORL\\WinVNC3\\Password
or
\\HKEY_USERS\\.DEFAULT\\Software\\ORL\\WinVNC3\\Password

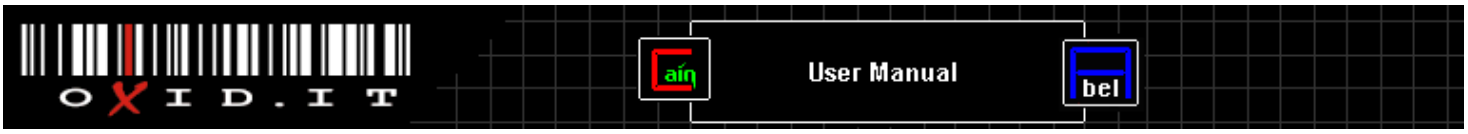


This feature decodes encrypted VNC passwords into their clear text form.



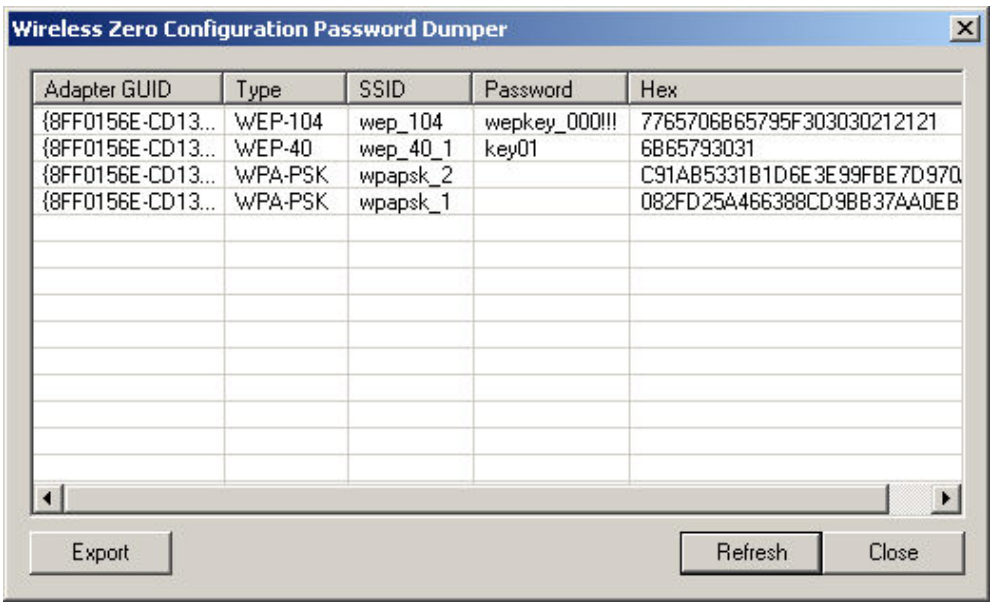
Usage

Simply cut and paste the encrypted password from the registry into the dialog; the decoder will do the rest.



Wireless Zero Configuration Password Dumper

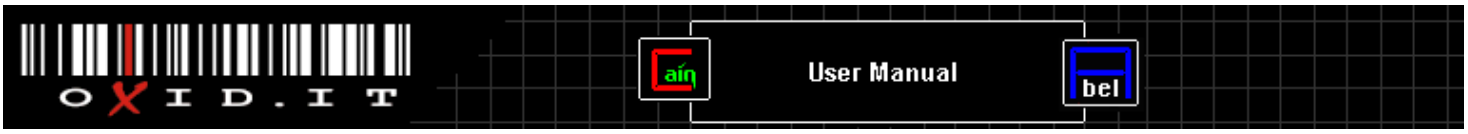
This feature enables the recovery of wireless keys stored by Windows's Wireless Zero Configuration Service.



WEP keys are automatically decoded. WPA-PSK keys are displayed in their hexadecimal format because of the usage of the SHA1 one-way hashing function.

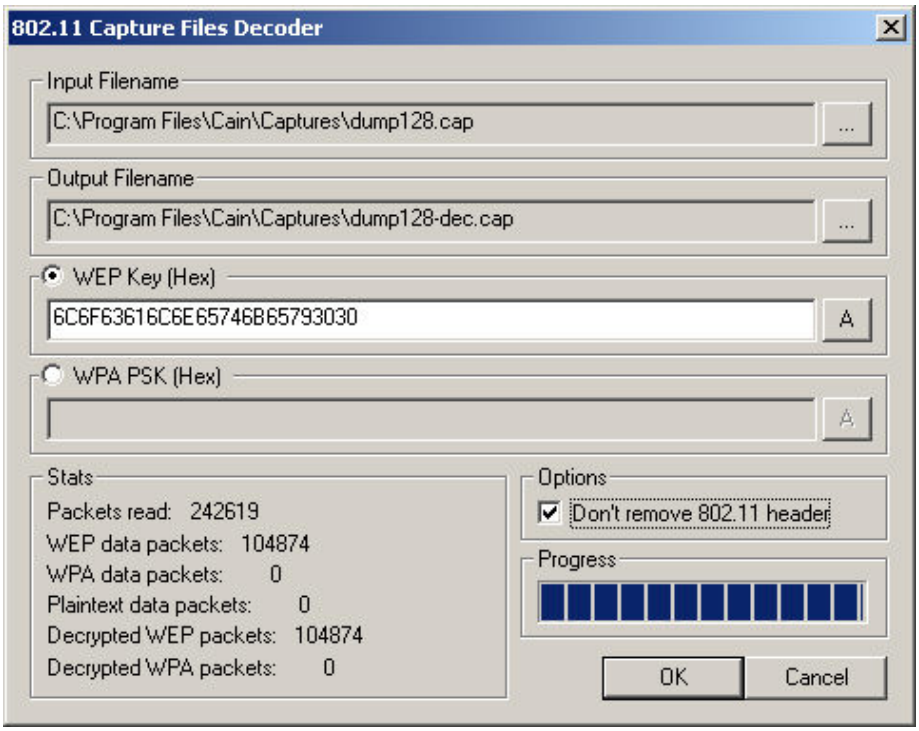
Usage

Wireless Zero Configuration Password Dumper dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button.



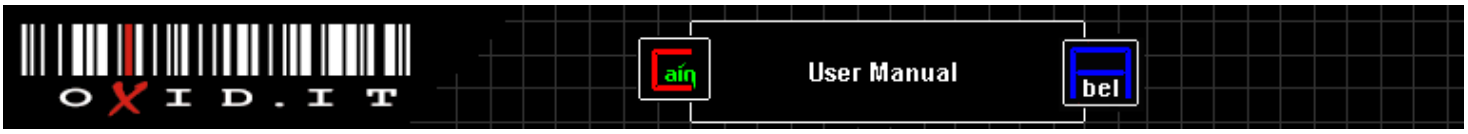
802.11 Capture Files Decoder

This feature covers the same functionality provided by the software airdecap from Christophe Devine. It can decode wireless capture files from Wireshark and/or Airodump-ng containing WEP or WPA-PSK encrypted 802.11 frames.



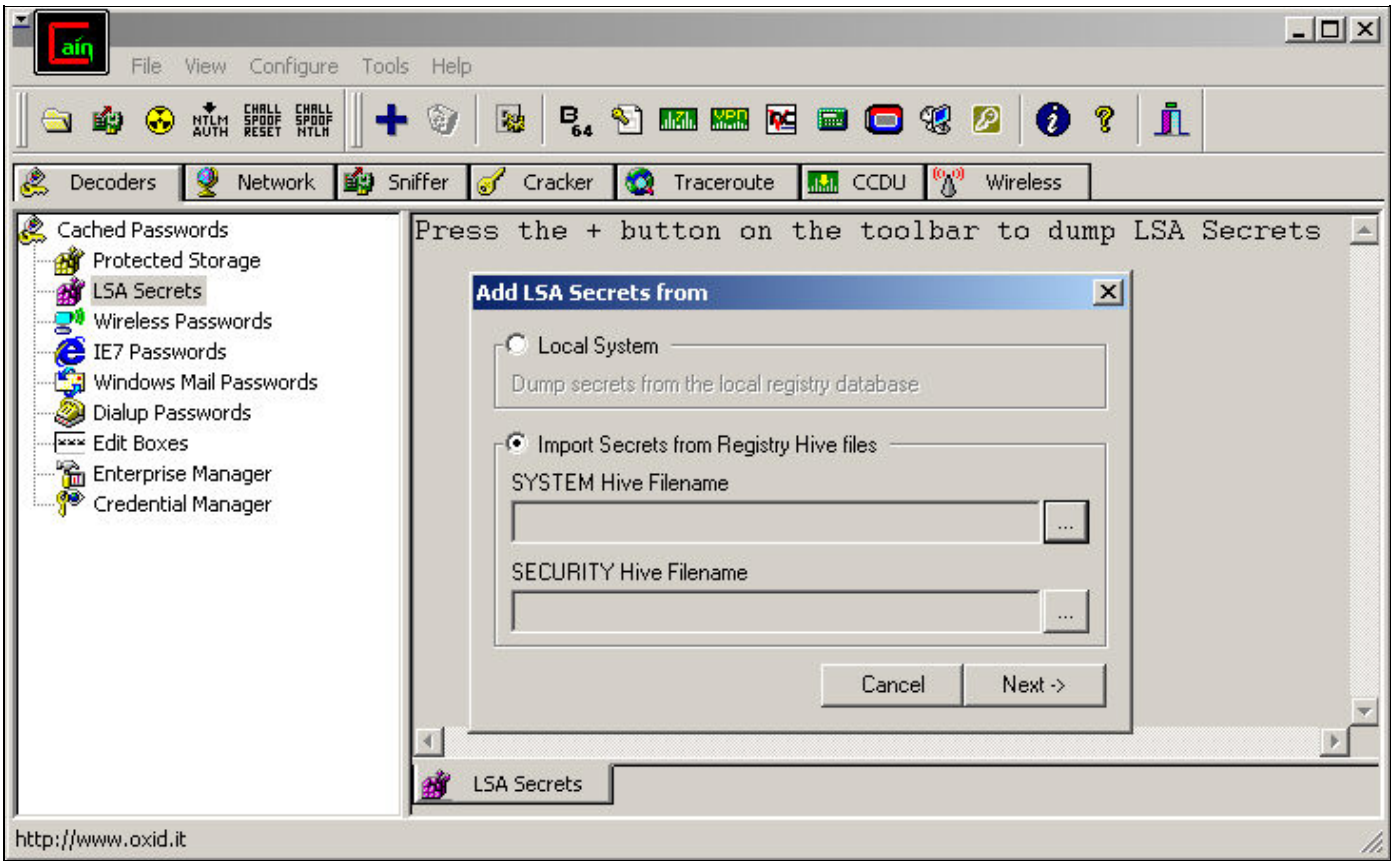
Usage

The 802.11 Capture Files Decoder can be invoked from the "802.11 Captures" list of the "Crackers" tab using the pop-up menu function "Decode". In order to successfully decode encrypted wireless frames you must provide the correct WEP or WPA-PSK encryption key.



LSA Secrets Dumper

LSA Secrets are used to store information such as the passwords for service accounts used to start services under an account other than local System. Dial-Up credentials and other application defined passwords also reside here.



How it works

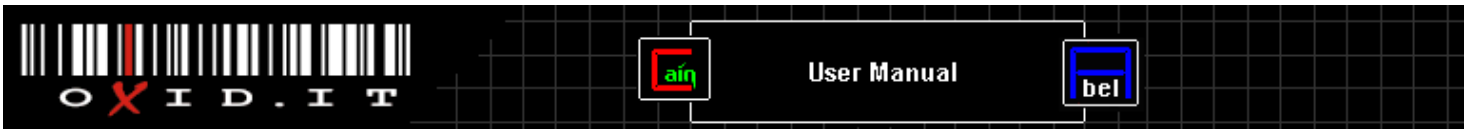
This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to decrypt LSA secrets present on the system. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed the thread will run with the same access privileges of the Local Security Authority Subsystem; it will load the function "DumpLsa" from Abel.dll which will open and query each secret using the LsarOpenSecret and LsarQuerySecret APIs from LSASRV.DLL. The thread stores the data returned from these functions in a temporary file named lsa.txt located in the same directory of the program. Finally, the content of this file is put on the screen and the temporary file is deleted.

Usage

To dump the content of the LSA Secrets you can press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. You can choose to dump LSA secrets from the local system or from external registry hive files (SYSTEM and SECURITY); Windows Vista hive files are also supported.

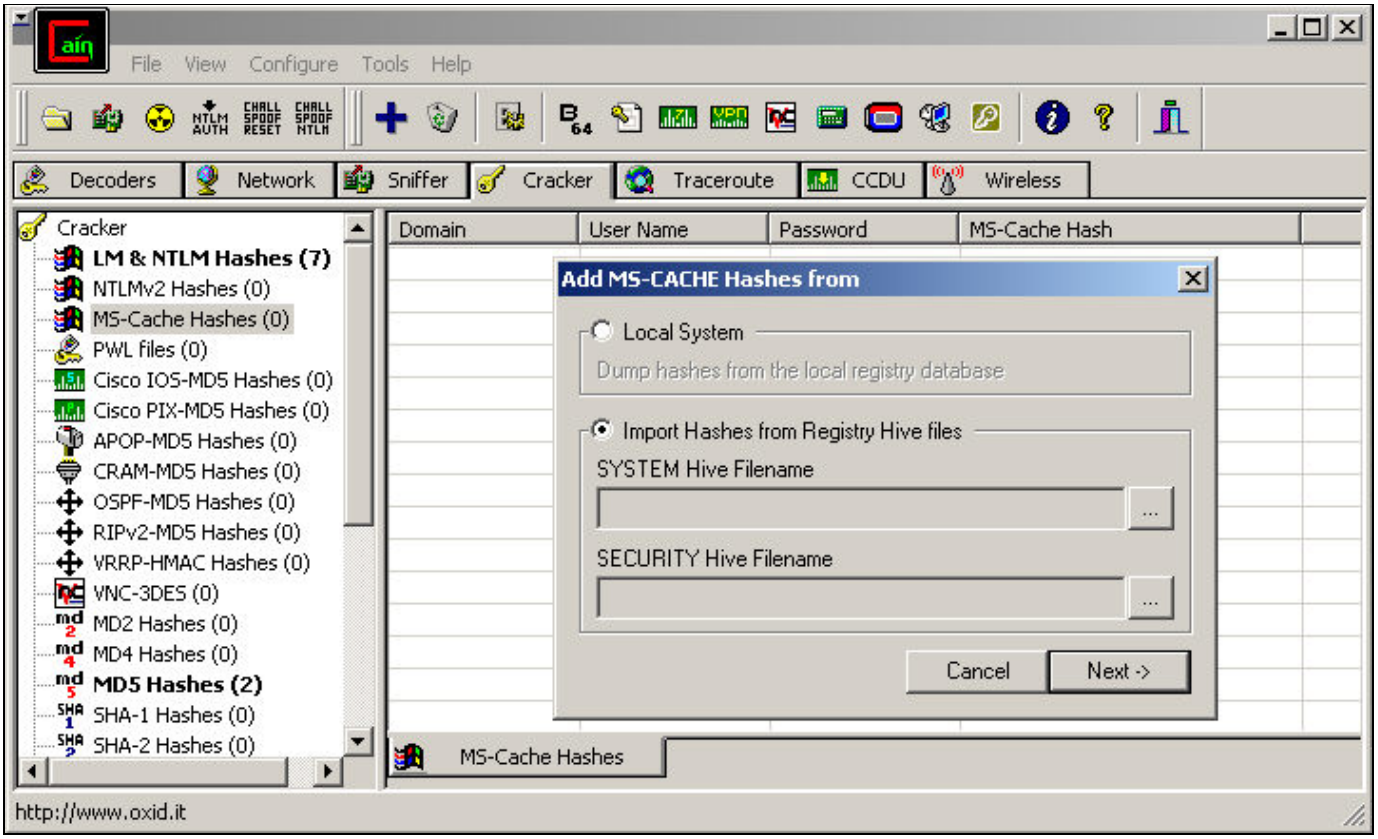
Requirements

This feature requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Abel.dll is also required by the remote thread injected into LSASS process.



MSCACHE Hashes Dumper

You should be familiar with the well known tool "CacheDump" from Arnaud Pilon; it is an application which dumps the cached password hashes stored into the registry. Cain's MSCACHE Hashes Dumper does exactly the same thing and allows you to import password hashes directly into the relative "MSCACHE Hashes" password cracker tab. By default, Windows stores a copy of domain logon passwords into the local registry; this enables the user to logon locally even if the domain controller is off-line or unavailable. Cached passwords are stored locally under the form of hashes encrypted with the NL$KM LSA secret. This feature decrypts cached hashes and prepares them to be cracked using Dictionary or Brute-Force attacks.



How it works

This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to extract MSCACHE hashes present on the system. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed the thread will run with the same access privileges of the Local Security Authority Subsystem; it will load the function "DumpCache" from Abel.dll that will
retrieve the NL$KM LSA secret and decrypt MSCACHE hashes stored in the registry. Unlike "CacheDump", the NL$KM LSA secret is obtained using the LsarOpenSecret and LsarQuerySecret APIs from LSASRV.DLL. The thread stores the data returned from these functions in a temporary file named cache.txt located in the same directory of the program. Finally, unencrypted MSCACHE hashes are put on the screen and the temporary file deleted.

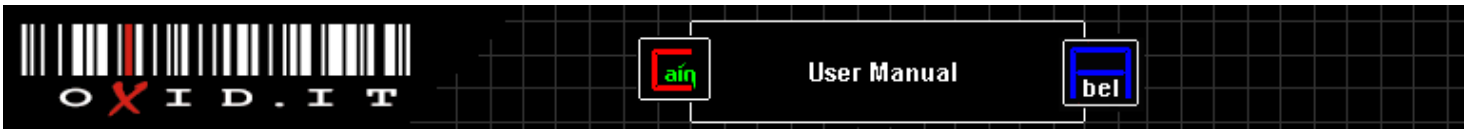
Usage

To dump MSCACHE hashes you can press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. Once dumped, cached password hashes can be sent to MSCACHE cracker using the list pop up menu. You can choose to dump MSCACHE hashes from the local system or from external registry hive files (SYSTEM and SECURITY); Windows Vista hive files

are also supported.

Requirements

This feature requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Abel.dll is also required by the remote thread injected into LSASS process.



MySQL Passwords Extractor

MySQL stores account's credentials into the USER table. Passwords are encrypted and saved under the form of hashes. This feature connects to the MySQL server via ODBC and dumps all account''s hashes into the MySQL Hashes Cracker list.

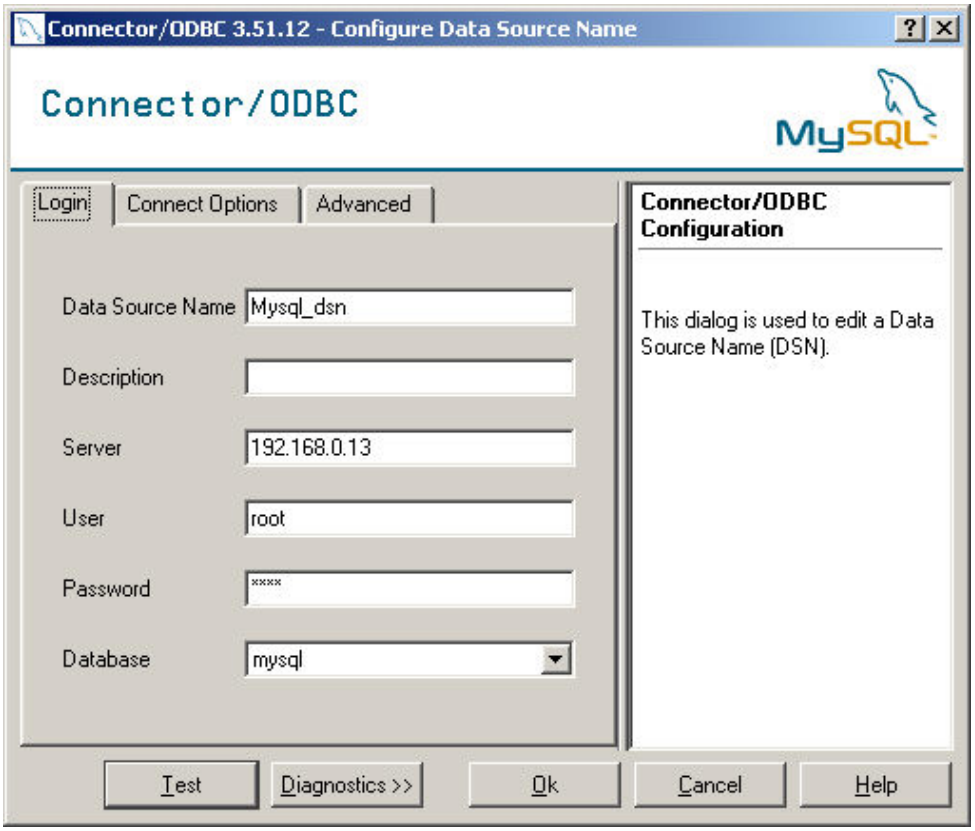
How it works

It connects to the server via ODBC and performs the following SQL command:

select user, password from user

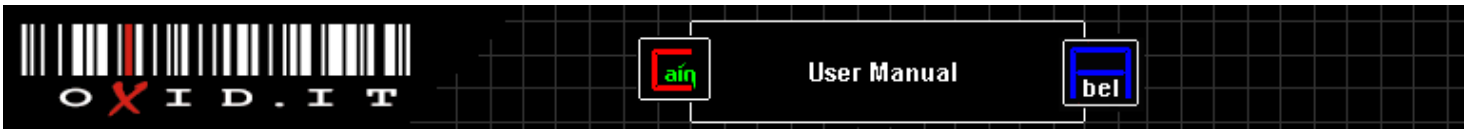
Usage

To dump the hashes go to the MySQL Hashes Cracker and press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. Choose the Data Source Name (DSN) for the target server and provide administrative credentials.



Requirements

This feature requires administrative privileges on the target database server and MySQL ODBC client installed.



NT Hashes Dumper

You should be familiar with the well known tool PWDUMP2 from Todd Sabin; it is an application which dumps the password hashes (OWFs) from NT's SAM (Security Account Manager) database, whether or not SYSKEY is enabled on the system. Cain's NT Hashes Dumper does exactly the same thing and allows you to import password hashes directly into the relative "LM & NTLM Hashes" password cracker tab.
What Cain's NT Hashes Dumper offers more than PWDUMP2 is the ability to dump password history hashes. Windows can be instructed to remember a number of previous user's passwords using the Password Security Policy "Enforce Password History". In this way the user cannot choose a password used before as the new one. The operating system stores history passwords under the same form as those currently used but those kind of hashes are not returned, as in PWDUMP2, by the "SamrQueryInformationUser" function of SAMSRV.DLL; they have to be extracted using the native function "SamIGetPrivateData" and decrypted later by "SystemFunction025" and "SystemFunction027" of ADVAPI32.DLL.

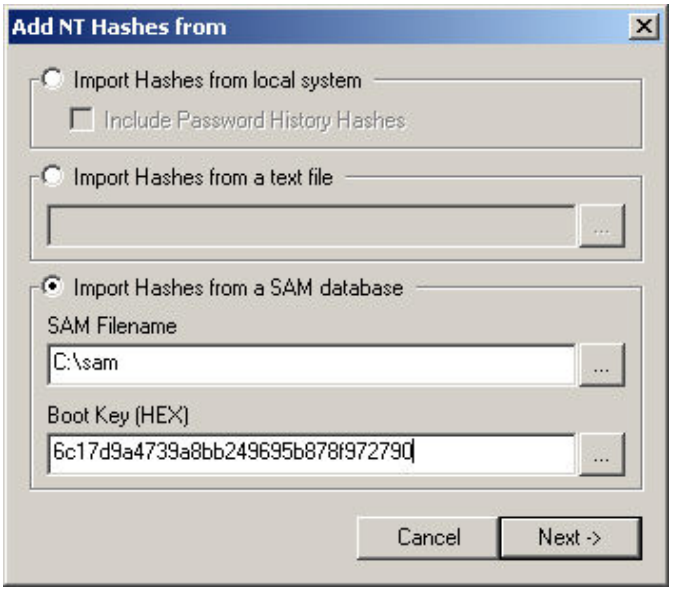
How it works

This feature of the program follows the same methodology used by Todd Sabin in his PWDUMP2 program to dump NT hashes present on the system. It uses the "DLL injection" technique to run a thread in the same security context of the Local Security Authority Subsystem process. The thread's executable code must first be copied to the address space of LSASS process and this requires an account with the SeDebugPrivilege user right. By default only Administrators have this right.
Once injected and executed the thread will run with the same access privileges of the Local Security Authority Subsystem; it will load the functions "DumpHashes"and "DumpHistory" from Abel.dll that will enumerate user's hashes present in the SAM database. This is done by mean of some native functions of SAMSRV.DLL library like "SamrEnumerateUsersInDomain", "SamrOpenUser", "SamrQueryInformationUser" and "SamIGetPrivateData". The thread stores the data returned from these functions in two temporary files named hashes.txt and history.txt located in the same directory of the program. Finally, the content of these files is put on the screen and the temporary files are deleted.

Cain can also import SYSKEY encrypted NT hashes from "off-line" SAM database files. This feature requires the correct Boot Key (Startup Key), created with the SYSKEY utility, to decrypt the encrypted hashes. The Boot Key is usually stored in the SYSTEM registry file, you can use Cain's Syskey Decoder to recover it for you.

Usage

To dump NT hashes you can press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar.



From the dialog you can choose the source of the import function, the local system, a text file (from PWDUMP or L0phtCrack) or an

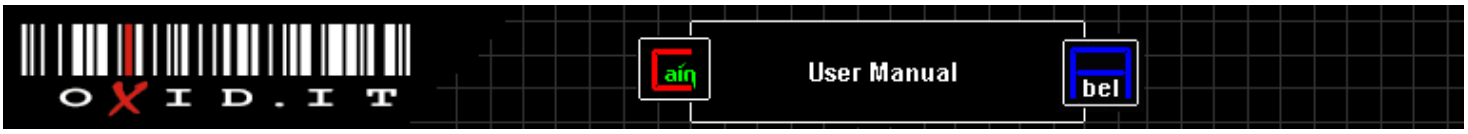
off-line SAM file.

If you need to recover hashes from a SAM file not encrypted by SYSKEY, simply leave the Boot Key field empty.

Once dumped, password hashes can be sent to LM & NTLM cracker using the list pop up menu.

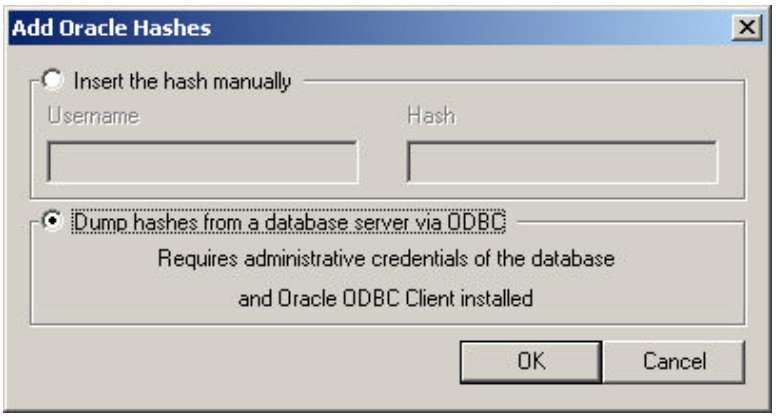
Requirements

The local system import function requires an account with the SeDebugPrivilege user right. By default only Administrators have this right. Abel.dll is also required by the remote thread injected into LSASS process. The extraction from a SYSKEY encrypted SAM file requires the correct Boot Key to decode the hashes.



Oracle Passwords Extractor

Oracle stores account's credentials in DBA_USERS / SYS.USER$ tables. Passwords are encrypted with DES in CBC mode and saved under the form of hashes. This feature connects to the Oracle server via ODBC and dumps all account"s hashes into the Oracle Hashes Cracker list.



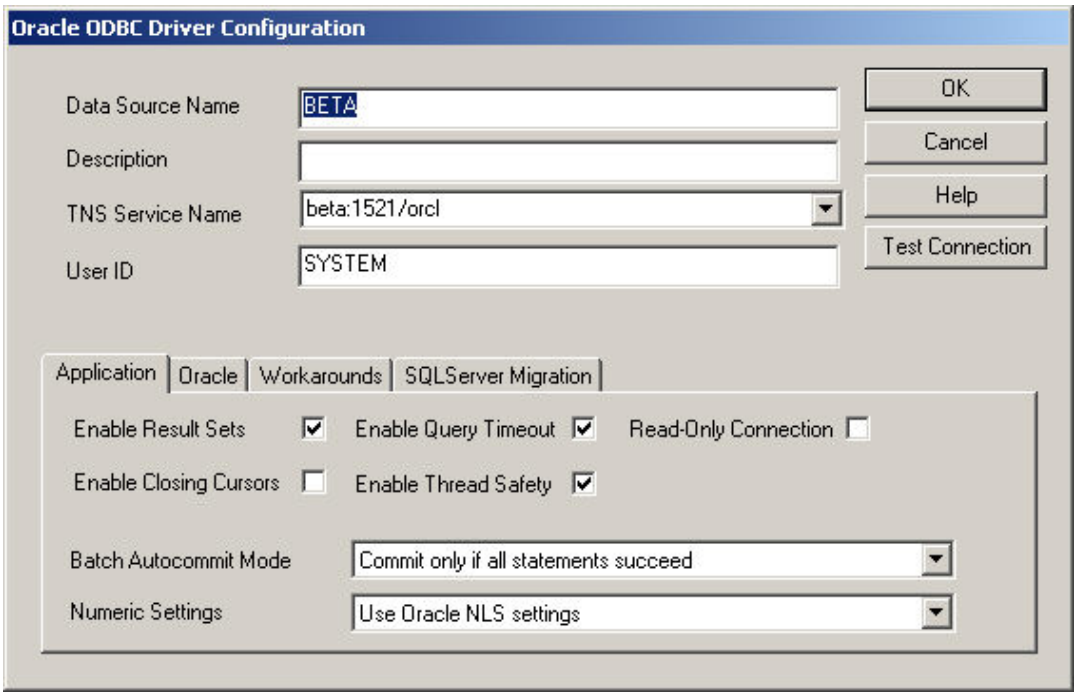
How it works

It connects to the server via ODBC and performs the following SQL command:

select username, password from DBA_USERS
select name,password from SYS.USER$

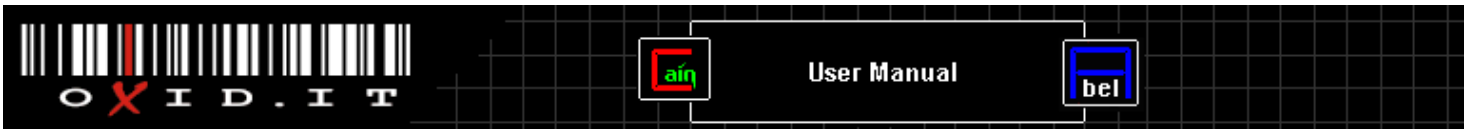
Usage

To dump the hashes go to the Oracle Hashes Cracker and press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. Choose the Data Source Name (DSN) for the target server and provide administrative credentials.



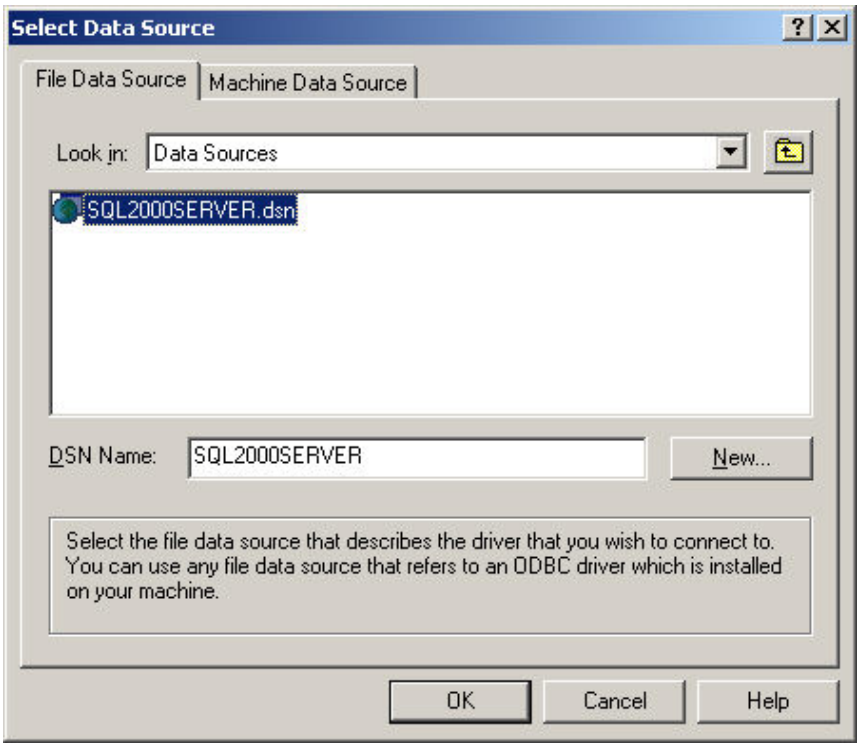
Requirements

This feature requires administrative privileges on the target database server and Oracle ODBC client installed.



SQL Server 2000 Password Extractor

Microsoft SQL Server 2000 stores the credentials of its accounts in the "**master**" database. User's passwords are encrypted under the form of salted SHA-1 hashes into the table "**sysxlogins**". This feature connects to the server using ODBC and dumps all SQL user's hashes into the MSSQL Hashes Cracker list.



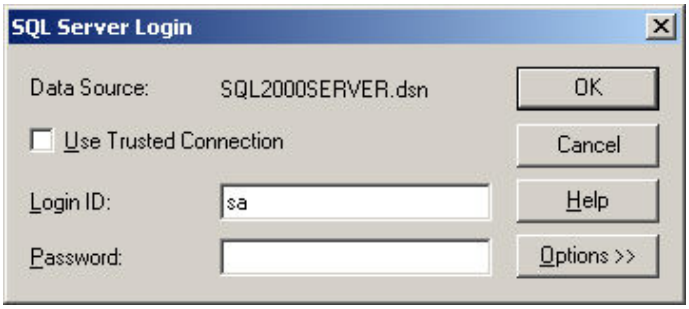
How it works

It connects to the server via ODBC and performs the following SQL command:

select name, password from master..sysxlogins

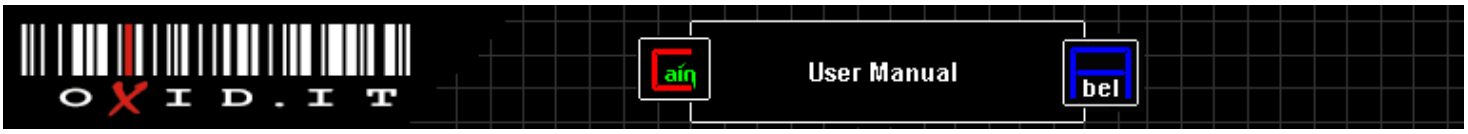
Usage

To dump the hashes go to the MSSQL Hashes Cracker and press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. Choose the Data Source Name (DSN) for the target server and provide system administrator (SA) credentials.



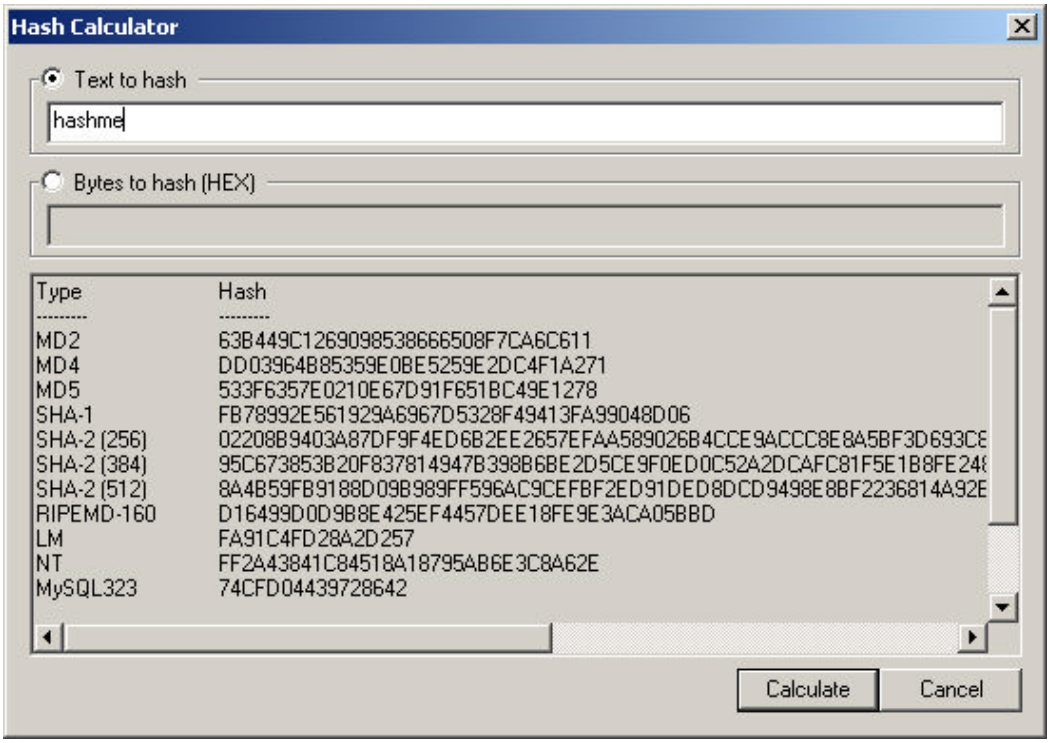
Requirements

This feature requires SQL Administrator's privileges on the target database server.



Hash Calculator

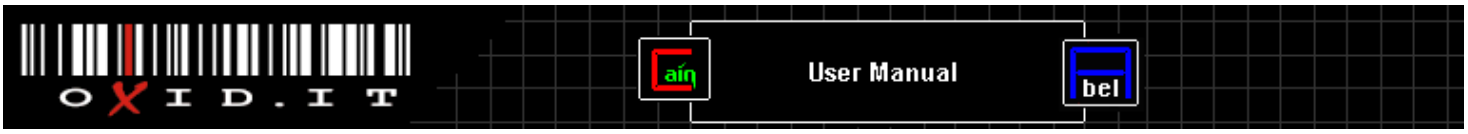
Cain's Hash Calculator produces hash values from the input text.



It supports most common hashing algorithms like MD2, MD4, MD5, SHA-1 ... plus some other specific ones like LM, NTLM, MySQL323 and Cisco PIX.

Usage

The Hash Calculator dialog can be activated from the main menu under "Tools" or pressing the relative toolbar button. Write the text you want to hash into the dialog and press the "Calculate" button.

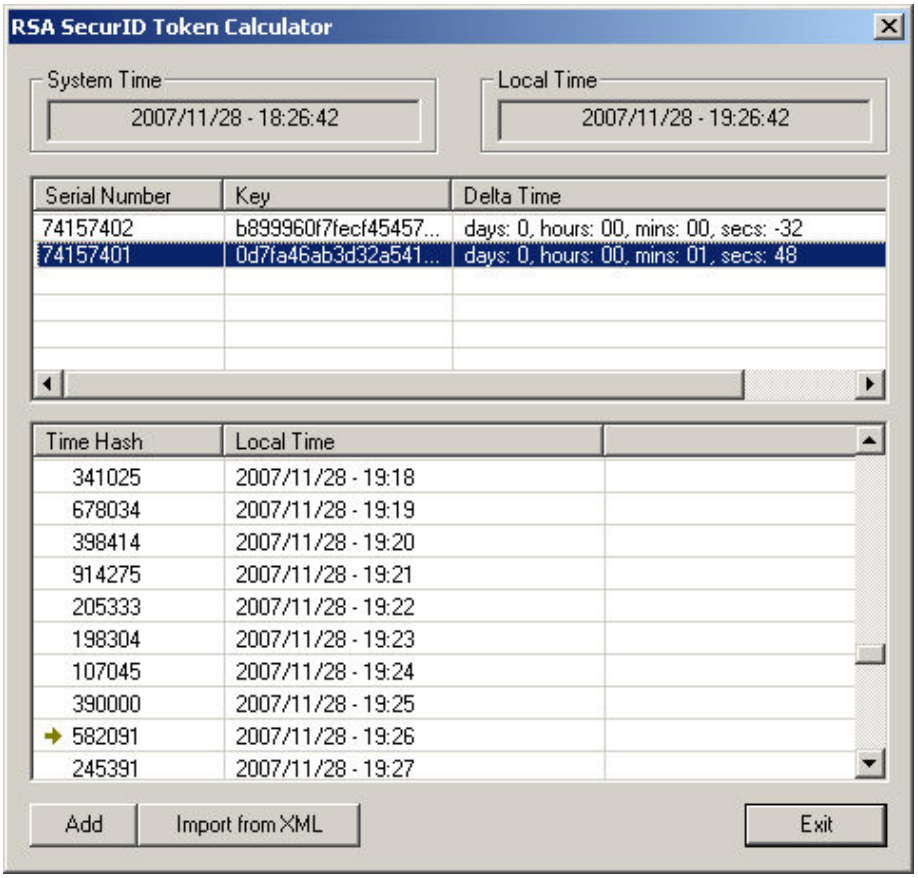


RSA SecurID Token Calculator

Have you ever seen it ?

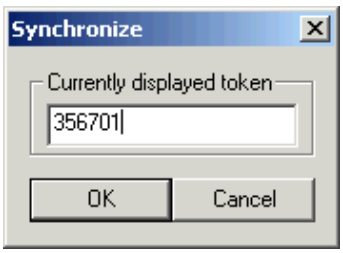
The RSA SecurID Key Fob is a lightweight and water-resistant token that is well suited for a variety of environments beyond the corporate office. Network users are today accessing enterprise networks from virtually any remote location imaginable, whether it be from home, a hotel room, an airport terminal or even outdoor locations. The SecurID Key Fob displays a randomly generated access code, which changes every 60 seconds. The SecurID Key Fob provides two-factor authentication: the user logs in by entering a secret personal identification number (PIN) followed by the current code displayed on the SecurID token.

Cain's RSA SecurID Token Calculator can generate the numbers displayed on the token before they appears. The token generation algorithm uses essentially two parameters: the key fob serial number and a token activation key; each of them are usually provided by the vendor in *.XML files.



Time Syncronization

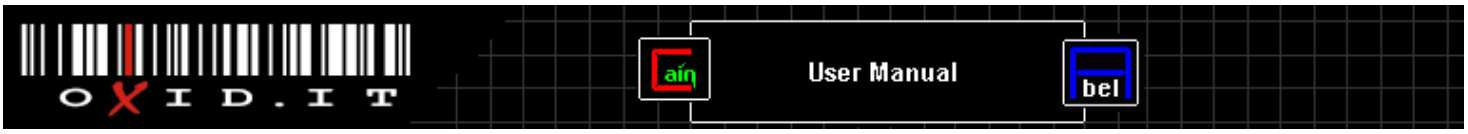
The token values are time hashes calculated every two minutes and displayed every 60 seconds. If you system time is not synchronized with the key fob internal time, calculated tokens will be correct but not valid to be used; for this reason the program must consider the delta time between the two clocks. Time synchronization can be performed for each token by mean of the upper list pop up menu:



The synchronization dialog requires the user to insert the number that is currently displayed on the token.

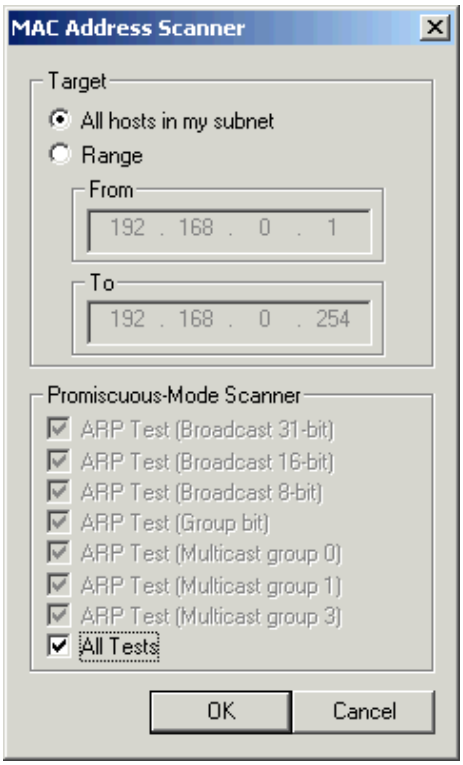
Usage

Use the dialog to import the token's parameters from *.XML file or manually enter the serial number and activation key (token seed) of the desired key fob; then click on the serial number in the upper list to generate tokens.



Promiscuous-mode Scanner

The Promiscuous-mode scanner allows you to identify sniffers and network Intrusion Detection systems present on the LAN. It implements the recognition method explained in the paper "**Promiscuous node detection using ARP packets**" by Daiji Sanai at The Black Hat Briefings 2001. This feature is included in the MAC Scanner and relies on responses received from various tests based on ARP packets.



It is possible to select the test to perform from the MAC Scanner dialog; positive results are reported into the "Hosts" list with an * in the relative column.
Be warned that not all operating systems respond in the same way; an example of the results from a Windows machine follows:

Network card not in promiscuous-mode (not sniffing)

IP address	MAC address	OUI fingerprint	Host name	B31	B16	B8	Gr	M0	M1	M3
192.168.0.10	00C026880898	LANS TECHNOLOGY CO., LTD.			*				*	

Network card into promiscuous-mode (sniffing)

IP address	MAC address	OUI fingerprint	Host name	B31	B16	B8	Gr	M0	M1	M3
192.168.0.10	00C026880898	LANS TECHNOLOGY CO., LTD.		*	*				*	

As you can see Windows machines, that are not sniffing the network, normally respond to ARP Test (Broadcast 16-bit) and ARP Test (Multicast group1) only. On the contrary when a sniffer is activated, and the network card is put into promiscuous-mode, they start to respond at ARP Test (Broadcast 31-bit) as well.

Prerequisites

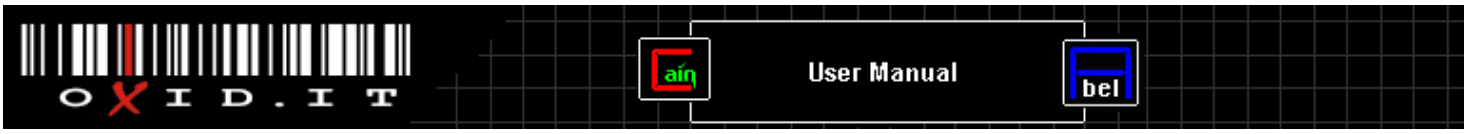
The sniffer must be activated.

Limitations

Because of the use of ARP packets, that cannot cross routers or VLANs, this feature works only inside your broadcast domain.

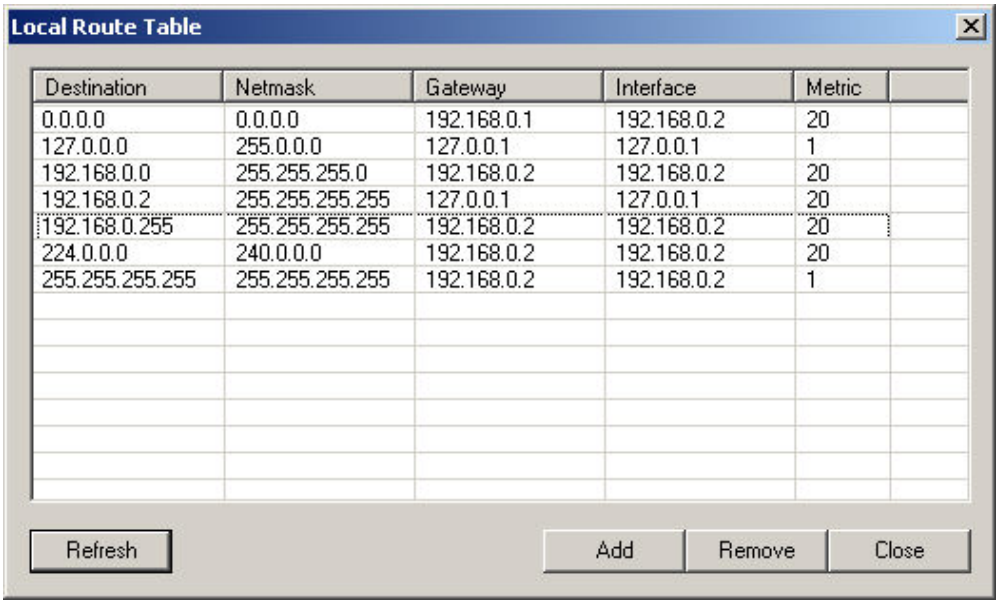
Usage

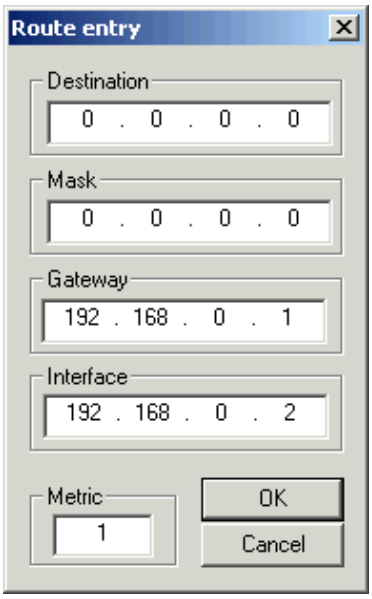
The promiscuous-mode scanner is activated using the MAC Scanner dialog.



Route Table Manager

Cain's Route Table manager covers the same functionality offered by the Windows tool "route.exe".

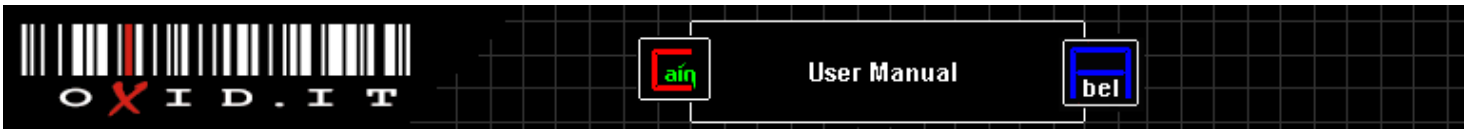




This utility enumerates all entries present in the local Windows route table. It also enables you to add new routes and modify/remove existing ones.

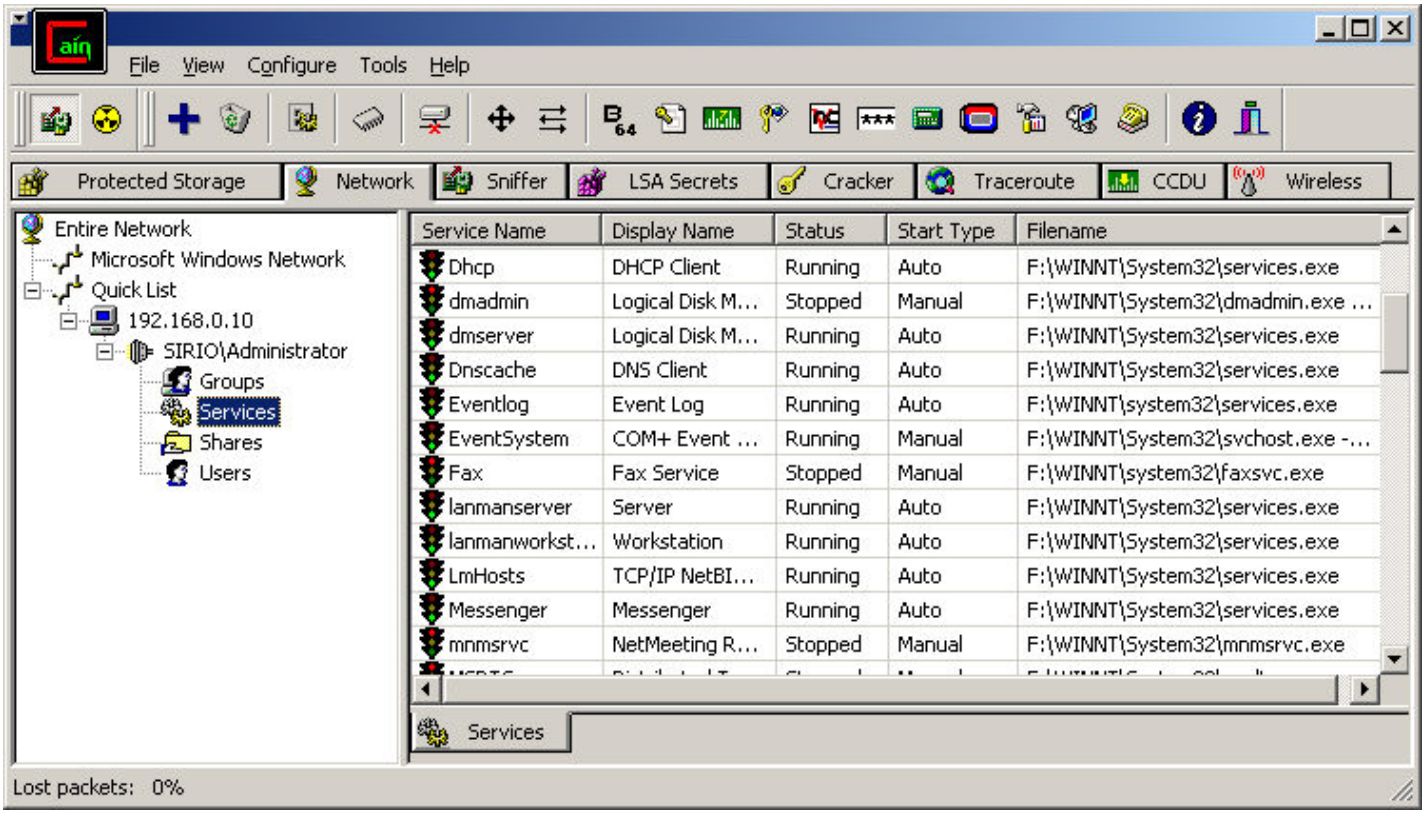
Usage

The utility can be activated using the relative toolbar button.



Service Manager

This feature allows you to start, stop, pause/continue or remove an NT service on a Windows machine.

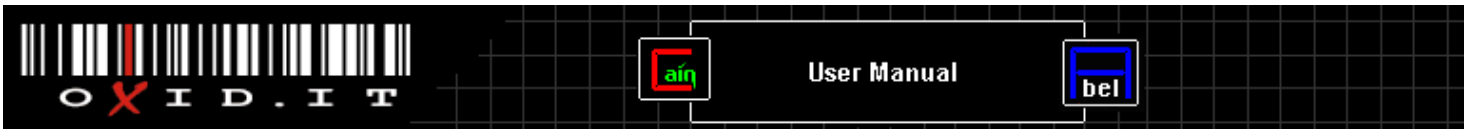


Usage

You can enumerate local/remote services from the left tree in the Network tab. To control a service right click on the list and choose the function within the pop up menu.

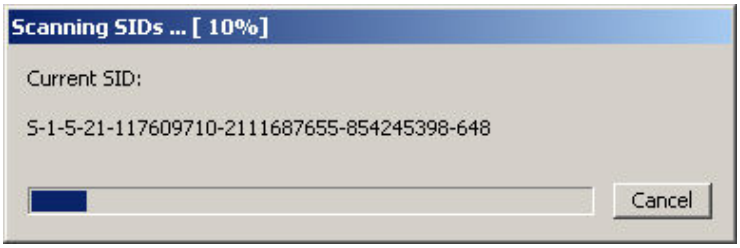
Requirements

This feature requires Administrator's privileges on the target machine.



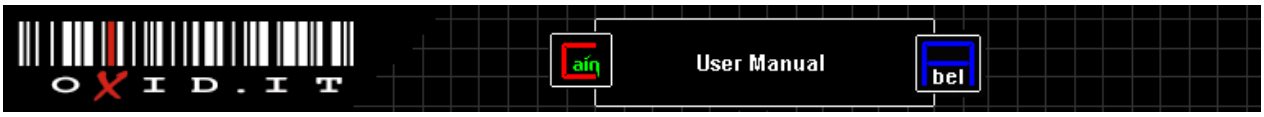
SID Scanner

Cain's SID Scanner allows the enumeration of users, by NULL Sessions, on systems the has the "**RestrictAnonymous**" parameter set to 1.
This parameter that resides under the registry key *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA* restricts the information available to anonymous logon users as described at the following link: http://support.microsoft.com/default.aspx?scid=KB;en-us;143474.
Evgenii B. Rudnyi with his well known tool **sid2user**, demonstrated the possibility for an anonymous login user to list the account names even if this kind of protection is activated. The SID Scanner uses the same methodology of this tool to extract this kind of information.



Usage

To activate the SID Scanner you have to right click on the "Users" item in the left tree and choose the relative function from the pop up menu.



Sniffer

Cain's sniffer is principally focused on the capture of passwords and authentication information travelling on the network. It should not be compared to professional tools like Observer, SnifferPro or Ethereal but unlike any other commercial protocol analyzer it has been developed to work on switched networks by mean of APR (Arp Poison Routing), another feature included in the program.

Protocol Filters

There is a BPF (Berkeley Packet Filter) hard-coded into the protocol driver that performs some initial traffic screening. The filter instructs the protocol driver to process only ARP and IP traffic; other protocols, like NetBEUI for example, are not processed.

Password Filters

The sniffer includes several password filters that can be enabled/disabled from the main configuration dialog; they are used to capture credentials from the following protocols:

Protocol	Authentication Types
FTP	Plaintext
HTTP / HTTP Proxy	Basic, Form, Cookie, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
IMAP	Plaintext, LOGIN, CRAM-MD5, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
POP3	Plaintext, APOP-MD5, CRAM-MD5, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
SMTP	Plaintext, LOGIN, CRAM-MD5, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
LDAP	Plaintext
NNTP	Plaintext, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
ICQ	v.7
VNC	3DES
TDS (Sybase / MSSQL)	v4.x, v5.0, v7.0 XOR, v7.0 NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
MySQL	v3.23, SHA-1
DCE/RPC	LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
SMB	Plaintext, LM, NTLMv1, NTLMv2, NTLM Session Security, NTLMSSP
MS Kerberos5	PreAuth encrypted timestamps
Radius	PreShared Keys, User's passwords
IKE	Aggressive Mode PreShared Keys (MD5 and SHA-1)
SNMP	Community strings
RIP	Plaintext, hash, RIPv2-MD5
HSRP	Plaintext
EIGRP	MD5
OSPF	Simple, MD5
VRRP	Simple, IP-AH, HMAC_MD5_96
SIP	Authorization and Proxy-Authorization (MD5)
GRE/PPP	PAP, CHAP-MD5, MS-CHAPv1, MS-CHAPv2
PPPoE	PAP, CHAP-MD5, MS-CHAPv1, MS-CHAPv2
Oracle TNS	DES, 3DES, AES-128, AES-192
Telnet	Entire session is dumped to file starting from a packet containing data to a FIN packet
*HTTPS	Entire session is dumped to file + the same types as HTTP
*LDAPS	Entire session is dumped to file + the same types as LDAP
*IMAPS	Entire session is dumped to file + the same types as IMAP
*POP3S	Entire session is dumped to file + the same types as POP3
*FTPS	Entire session is dumped to file + the same types as FTP
*SSH-1	Entire session dumped to file (FULL-DUPLEX, stealth, supports DES, 3DES, Blowfish symmetric encryption algorithms, auto-downgrade to SSH-1 if server version is v1.99)
MGCP/RTP	Captures and decode VoIP conversations that travels on RTP protocol and saves them as WAV files.
SIP/RTP	Captures and decode VoIP conversations that travels on RTP protocol and saves them as WAV files.

(*) = requires APR (Arp Poison Routing) to be enabled

Cain's sniffer filters are internally designed to survive into an unreliable world such as a network under ARP Poison attack; Cain uses different protocol state machines to extract from network packets all the information needed to recover the plain text form of a transmitted password. Some authentication protocols use a challenge-response mechanism, for this reason the sniffer needs parameters from each Client->Server and Server->Client traffic. On switched networks this can be achieved with a mirror port on the switch or if APR reaches the FULL-Routing state.

When APR (Arp Poison Routing) is enabled, the sniffer must process packets that normally aren't seen and also re-route them to the correct destination; this can cause performance bottlenecks on heavy traffic networks so be careful. APR's main advantage is that it enables sniffing on switched networks and also permits the analysis of encrypted protocols such as HTTPS and SSH-1.

Passwords and hashes are stored in .LST files in the program's directory. These files are **TAB separated files** so you can view or import them with your preferred word processor (e.g.: POP3.LST contains passwords and hashes sniffed from the POP3 protocol).

For HTTPS, SSH-1 and Telnet protocols entire sessions are decrypted and dumped into text files using this naming convention:

<Protocol name>-<Year><Month><Day><Hour><Minute><Second><Milliseconds>-<client port>.txt

(e.g.: Telnet-20041116135246796-1141.txt)

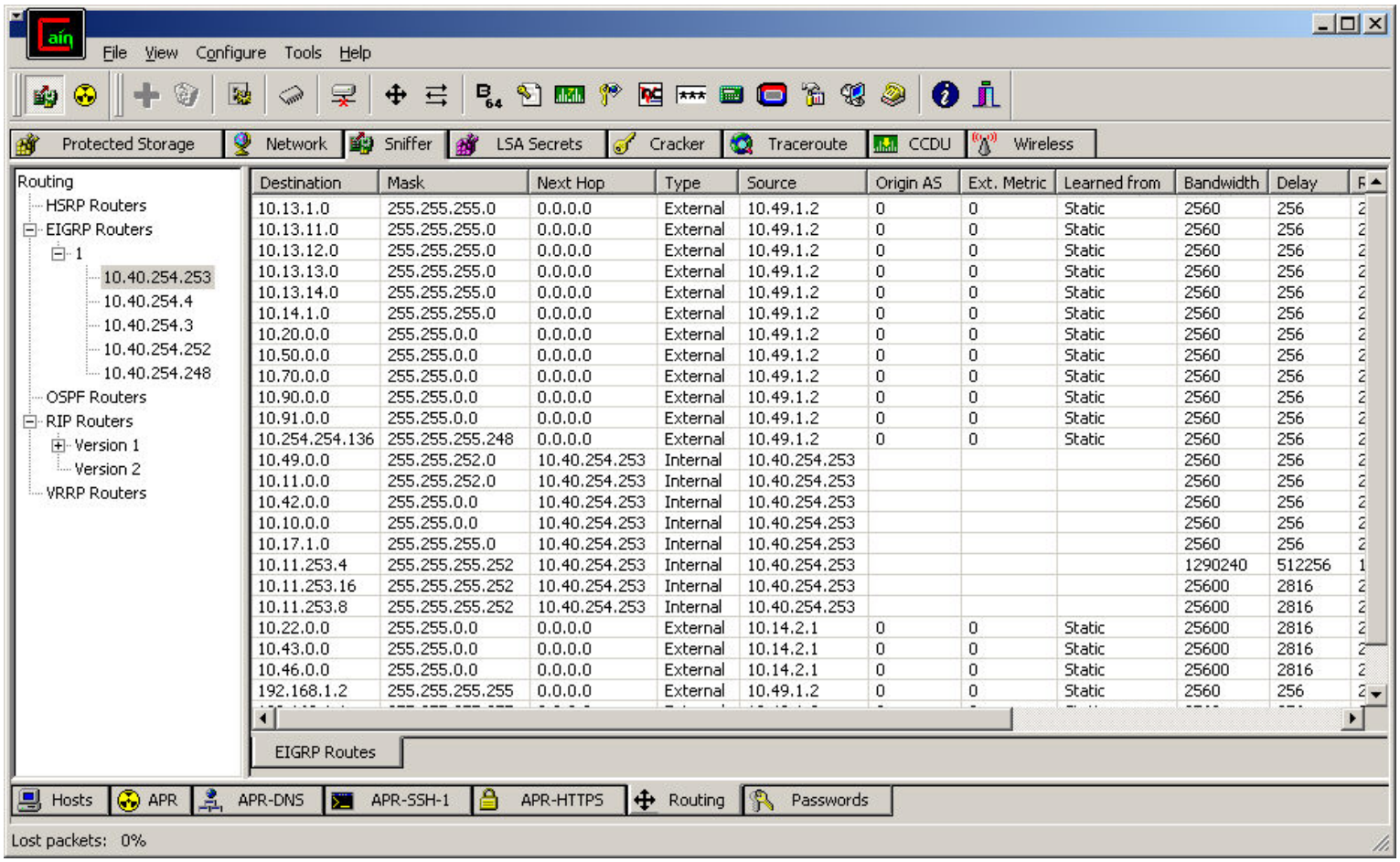
For more information on .LST files please refer to the Installation page.

Off-line capture file processing

The sniffer can also process file captures (from Ethereal, Tcpdump and Winpcap) in off-line mode. The captures can be imported using the "open file" button of the sniffer's toolbar; when processing network traffic off-line all APR's functions are automatically disabled.

Routing Protocols Analysis

Routing protocols like VRRP, HSRP, RIP, OSPF, EIGRP are also analyzed by the program. This enables a quick identification of the subnet routing and perimeter.



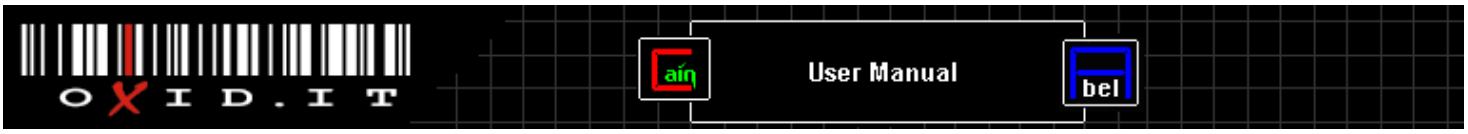
For EIGRP and RIP protocols, the "Routes Extractor" feature will also dump the actual routing table shared between routers. The feature is only supported if these protocols don't require authentication.

Usage

The sniffer is activated/deactivated using the relative toolbar button and its parameters can be configured from the main configuration dialog.

Requirements

- Supported Ethernet network adapter
- Winpcap Packet Driver (v2.3 or above) from Politecnico di Torino.



APR

APR (ARP Poison Routing) is a main feature of the program. It enables sniffing on switched networks and the hijacking of IP traffic between hosts. The name "ARP Poison Routing" derives from the two steps needed to perform such unusual network sniffing: an ARP Poison Attack and routing packets to the correct destination.

ARP Poison Attack

This kind of attack is based on the manipulation of host's ARP caches. On an Ethernet/IP network when two hosts want to communicate to each other they must know each others MAC addresses. The source host looks at its ARP table to see if there is a MAC address corresponding to the destination host IP address. If not, it broadcasts an ARP Request to the entire network asking the MAC of the destination host. Because this packet is sent in broadcast it will reach every host in a subnet however only the host with the IP address specified in the request will reply its MAC to the source host. On the contrary if the ARP-IP entry for the destination host is already present in the ARP cache of the source host, that entry will be used without generating ARP traffic.

Q: Now what happens if the source host has in its ARP cache an incorrect MAC address associated to the IP address of the destination host ?
A: Simple, it will start the communication with the destination host using the incorrect MAC address in Ethernet frames.

Q: And what happens if that incorrect MAC address corresponds to the MAC address of our network sniffer ?
A: The traffic will reach our sniffer even if every host is connected to a network switch forwarding frames on port basis.

Q: How can someone change the addresses contained in host's ARP caches ?
A: The Address Resolution Protocol (ARP) is a stateless protocol that does not require authentication so a simple ARP Reply packet sent to an host can force an update in its ARP cache.

Q: Can I use this kind of attack on the Internet ?
A: No. ARP protocol does not cross routers or VLANs so ARP Poison attacks are useless outside Level2 "Broadcast Domains".

Manipulating ARP caches of two hosts, it is possible to change the normal direction of traffic between them. This kind of traffic hijacking is the result of an ARP Poison attack and also a prerequisite to achieve a "Man-in-the-Middle" condition between victim hosts. The term Main-in-the-Middle refers to the fact that the traffic between hosts follows an obligated path through something before reaching the desired destination.

Re-Routing Packets

Now suppose that you successfully setup an ARP Poison attack between two hosts to intercept their network traffic. To do so you had specified the sniffer MAC address in ARP Poison packets and now you are forcing the two hosts to communicate through your computer.
In this situation the sniffer receives packets that are directed to its MAC address but not to its IP address so the protocol stack discards these packets causing a Denial of Service between the hosts. To avoid such problems the sniffer must be able to re-route poisoned packets to the correct destination. (You can't capture any password if hosts cannot communicate)

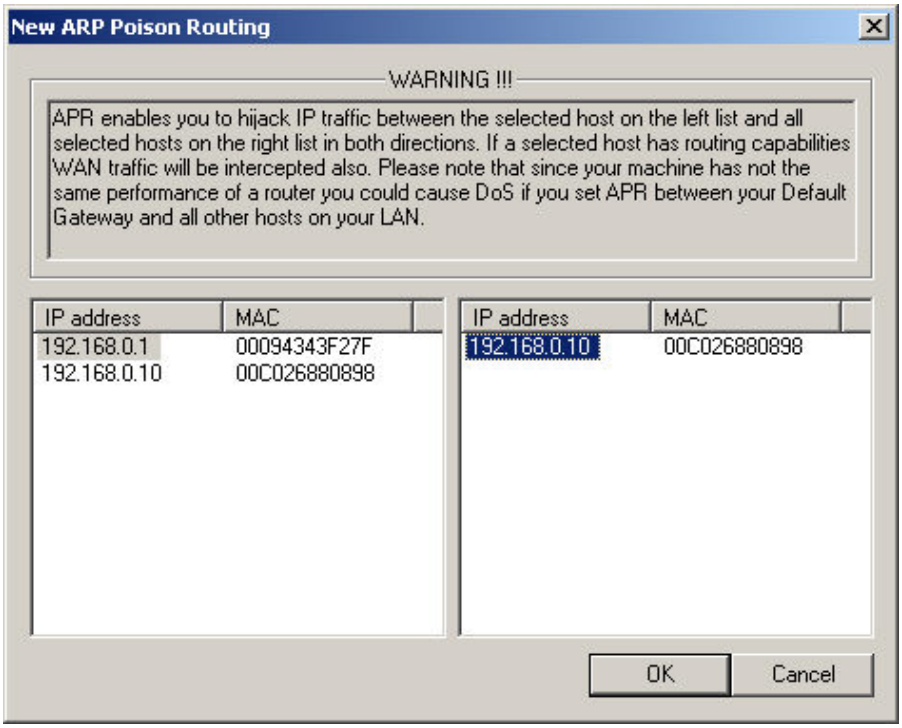
Prerequisites

In order to re-route poisoned packets to the correct destination, the program must know each IP-MAC association of victim hosts. This is why the user is asked to scan for MAC addresses first.

Configuration

This feature needs the configuration of some parameters that can be set from the configuration dialog. It is possible to specify a spoofed MAC and IP addresses to be used in ARP Poison packets; this makes it very difficult to trace back to the origin of the attack because attacker's real addresses are never sent across the network. On switched networks, the attack is also a stealth one from a central point of view because Cain's APR uses Unicast Ethernet destination addresses in ARP Poison packets; these packets will be routed by switches accordingly to their CAM tables and never sent in broadcast.

Victim hosts can be selected from the APR Tab using the + button in the toolbar:

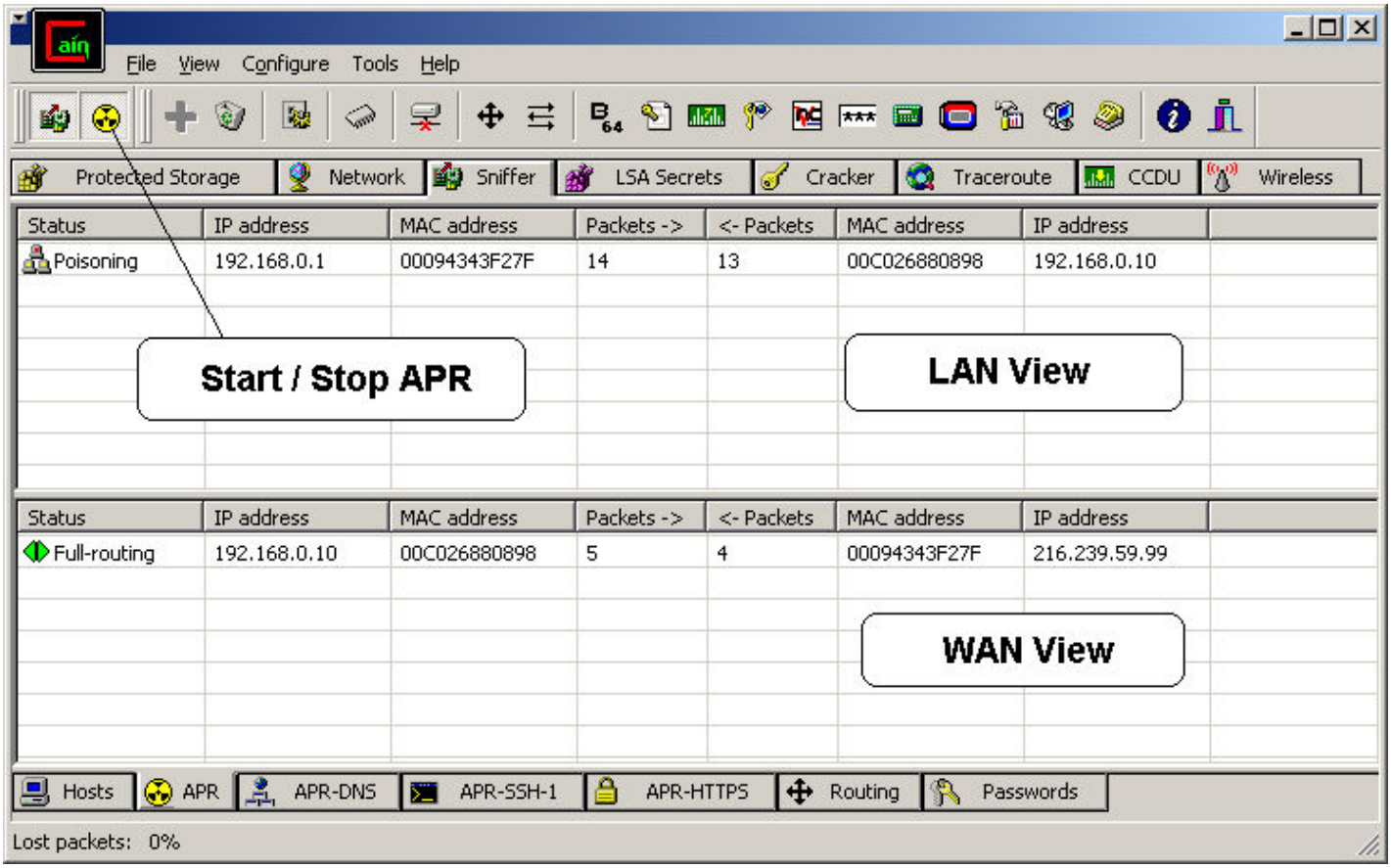


The meaning of this selection is: "I want to hijack all IP traffic that flows from host 192.168.0.1 and host 192.168.0.10 in each direction so that my workstation will be in a Man-in-the-Middle condition between them". In this way the program is configured to perform an ARP Poison attack directed to the selected hosts and at the same time the association needed to re-route poisoned packets is created. Cain's APR has been developed to handle attacks on multiple hosts at the same time so you can choose in the right list a pool of addresses.

The attack can now be enabled/disabled using the relative toolbar button; a separate thread will take care of sending ARP Poison packets at regular intervals as specified in the configuration dialog.

APR Views

You can monitor the traffic activity from the two views under the APR sub TAB. The upper view (LAN View) shows the number of re-routed packets between poisoned hosts and also the routing direction of the packets. It can happen that for some reason (static ARP entries for example) the attack is successful for one host only; in this case you will see the number of re-routed packets rising for one direction only meaning that the sniffer is processing half of the traffic expected.



The lower view (WAN View) shows the number of re-routed packets directed to or coming from an IP address which is external to the current subnet. If one of the two hosts is a router it is possible that Cain's APR will process WAN traffic too; in this case the lower list will be automatically populated with associations for WAN traffic.

When poisoning a router the following considerations arise:

- If you setup APR to hijack IP traffic between an internal host and its default gateway you will automatically intercept traffic from that host and all other hosts present in external networks connected by that gateway.

- When APR receives a packet originated from an internal host and directed to an IP address which is external to the current subnet it must re-route that packet to the correct gateway which is unknown.
The destination IP address present in the packet is the one of an external host and the destination Ethernet address is our sniffer MAC address..... the question arises as to where to re-route this packet if there are multiple exit point (gateways) in our LAN ? The packet could be sent in broadcast but this works only with routers, I checked that Checkpoint Firewalls for example discards packets directed to Unicast IP addresses encapsulated in frames with broadcast MAC addresses. This problem has been mitigated in the current version in this way: when APR does not know where to re-route packets it will use the best route found in the local operating system's route table.
If your LAN uses asymmetric routing you can modify the local route table using the Route Table Manager to avoid the above problem.

- Poisoning the subnet's default gateway with all other hosts in the LAN can cause traffic bottlenecks because APR does not have the same performance of an high speed router.

- Default gateways addresses are usually virtual addresses generated by HSRP or VRRP routing protocols. Consider if you are poisoning a normal host and the default gateway virtual address...
In this case a packet originated outside the local network and directed to an internal host will reach the sniffer but this packet could contain the real MAC address of the active HSRP / VRRP host as Ethernet source address. Because this source MAC address is not the one you setup in the APR list, the packet will not be re-routed by APR causing DoS. When you want to poison HSRP / VRRP virtual addresses you have to poison also real addresses of HSRP/ VRRP members.

APR WAN Status

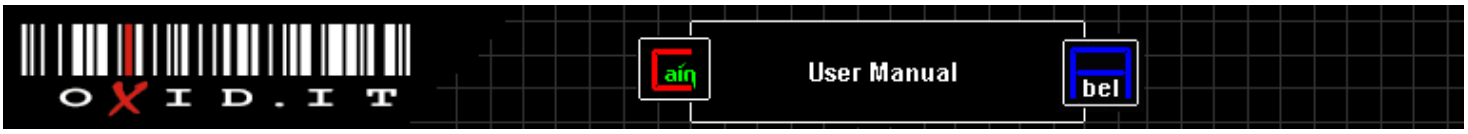
Each entry present in the WAN list can reach the following status:

- Broadcasting: This state means that APR received a packet from a host that resides on a different network and directed to an IP address of your broadcast domain. That packet must be routed by APR but the correct destination MAC address is not present in the host list. In this situation APR will broadcast that packet to all hosts in your LAN.

- Half-Routing: This state means that APR is routing the traffic correctly but only in one direction (ex: Client->Server or Server->Client). This can happen if one of the two hosts cannot be poisoned or if asymmetric routing is used on the LAN. In this state the sniffer looses all packets in an entire direction so it cannot grab authentications that use a challenge-response mechanism.

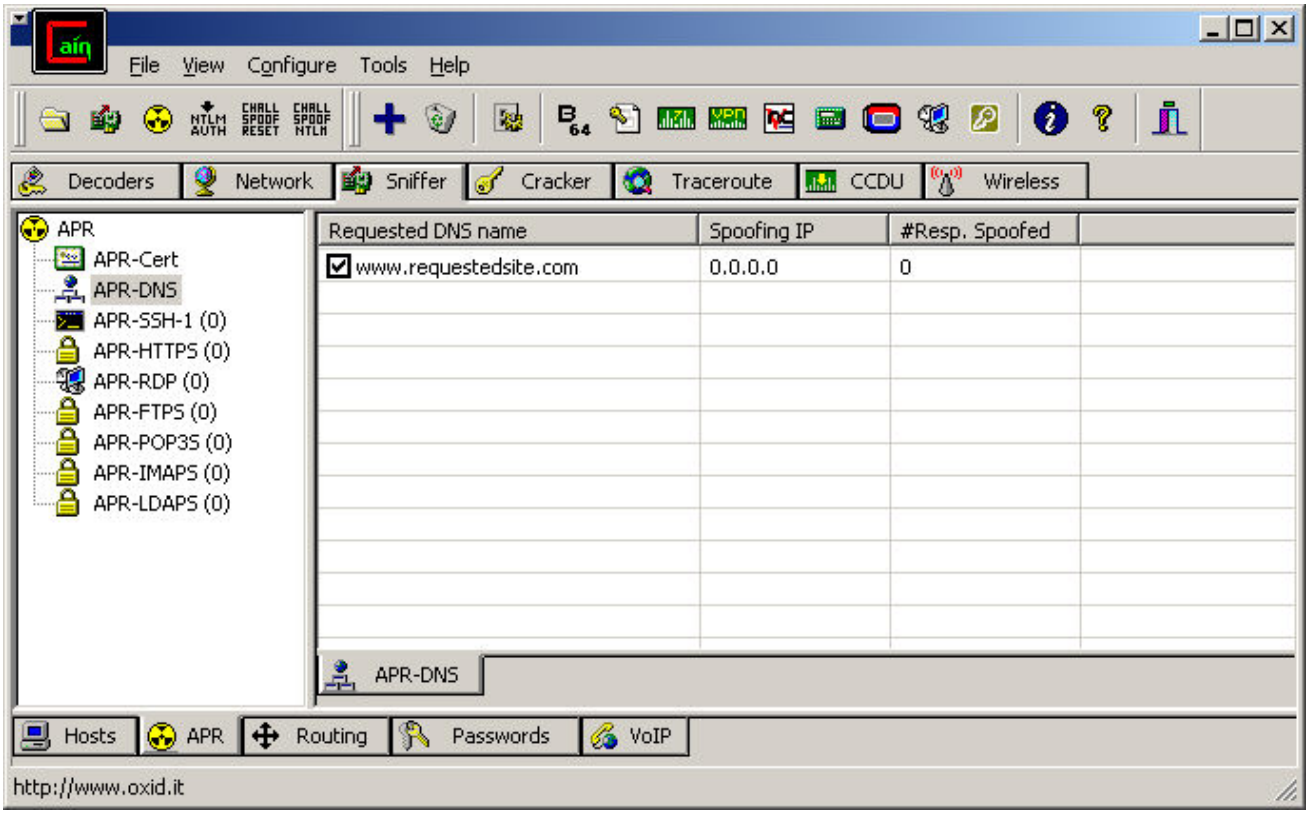
- Full-Routing: This state means that the IP traffic between two hosts has been completely hijacked and APR is working in FULL-DUPLEX. (e.g.: Server<->Client). The sniffer will grab authentication information accordingly to the filters set.

Notes
Where is the rest of the traffic if FULL-Routing is not reached ? Probably the immune host still uses the correct destination MAC address to reach the other host and its traffic does not flow through the sniffer machine. The same situation can happens if you are poisoning an HSRP Virtual address (usually a default gateway address) with a normal host in your LAN.



APR-DNS

This feature allows you to perform DNS spoofing attacks modifying DNS-Reply packets on the fly.



How it works

APR-DNS simply rewrites IP addresses in DNS-Reply packets. The sniffer extracts requested names from these packets and looks for an address association in the spoofing list. If there is a match, the packet is re-written using the relative spoofing IP address and then re-routed by APR engine. In this way the client that receives the spoofed reply is effectively redirected to the desired destination.

Note

Only Type-A (Host Address) fields in DNS Reply packets are rewritten by the program. DNS compressed names are also supported. From more information on DNS protocol you can take a look at DNS RFCs.

Prerequisites

This feature needs APR to be enabled and a Man-in-the-Middle condition between the DNS server and the victim host.

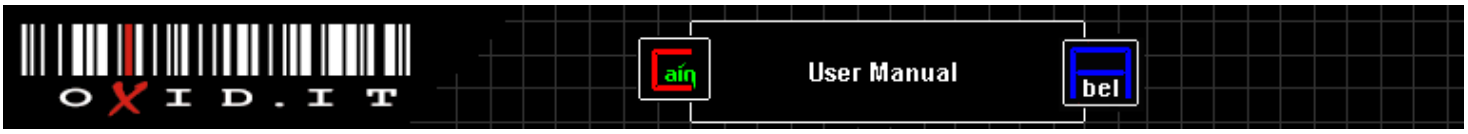
Limitations

Because of the use of the Winpcap driver this feature cannot be used to rewrite DNS Replies destined to the local host. The Winpcap driver cannot not stop packets before they enter the local protocol stack so legitimate replies are always received from the local machine.

Usage

The list of requested names and relative spoofing IP addresses to be rewritten must be configured first. To add an entry to the

APR-DNS list you can press the "Insert" button on the keyboard or click the icon with the blue + on the toolbar. You can enable/disable an entry using the check boxes in the first column. The "#Resp. Spoofed" column indicates the number of responses rewritten by the program for the specified DNS name.

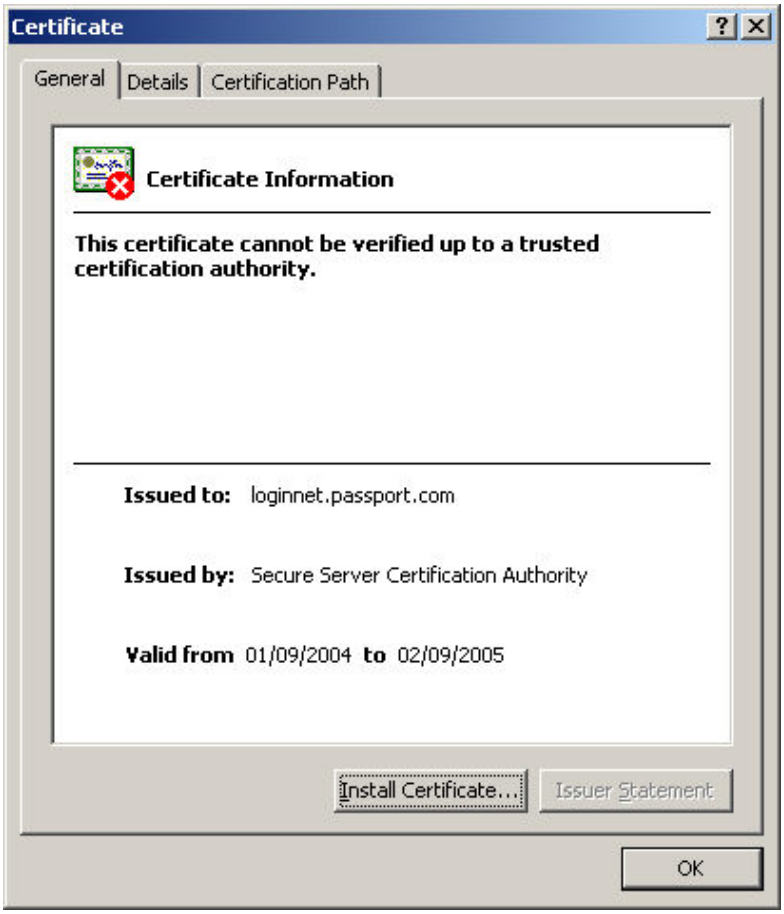


APR-HTTPS

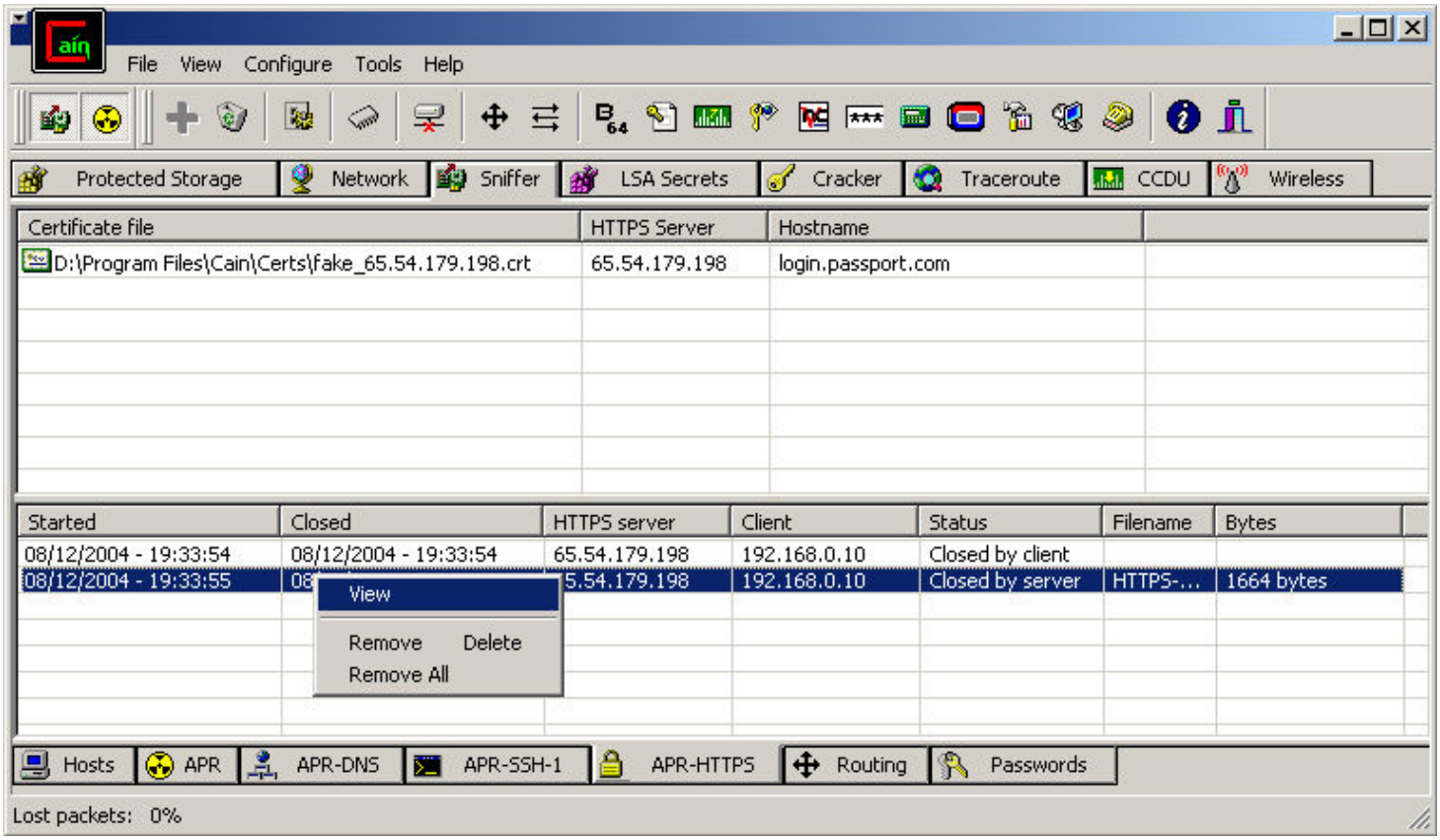
APR-HTTPS enables the capture and the decryption of HTTPS traffic between hosts. It works in conjunction with Cain's Certificate Collector to inject fake certificates into SSL sessions, previously hijacked by mean of APR. Using this trick it is possible to decrypt encrypted data before it arrives to the real destination performing a what so called Man-in-the-Middle attack.

Be warned that clients will notice this kind of attack because the server's certificate file injected into the SSL session is a fake one and although it is very similar to the real one it is not signed by a trusted certification authority. When the victim client starts a new HTTPS session, his browser shows a pop-up dialog warning about the problem.

APR-HTTPS uses the certificate files manipulated by the Certificate Collector. They contains the same parameters of the real ones except for asymmetric encryption keys; this deceives a lot of users to accept the server certificate and continue with the session.



The lower list in the APR-HTTPS tab contains all the session files that have been captured during the Man-in-the-Middle attack; decrypted data is saved in these text files located under the "HTTPS" subdirectory of the main installation folder.



How it works

Cain's HTTPS sniffer works in FULL-DUPLEX CLIENT-SIDE STEALTH mode; both server and client traffic is decrypted and if spoofing is enabled the attacker's IP and MAC addresses are never exposed to the victim client. Connections are accepted by a local "acceptor" socket listening on HTTPS port defined in the configuration dialog; this socket handle hijacked client connections but only when APR is enabled. OpenSSL libraries are used to manage SSL communications over two more sockets, one used for the traffic between the client <-> Cain and the other used for the traffic between Cain <-> server.

This is how all works step by step:

1) The HTTPS filter is enabled by the user in the configuration dialog
2) APR is enabled by the user using the button on the toolbar -> the Man-in-the-Middle attack is ready
3) The victim client starts a new session to an HTTPS enabled server (e.g. https://login.passport.com)
4) Packets from the client are hijacked by APR and captured by Cain's sniffer by mean of Winpcap driver
5) APR-HTTPS search for a fake certificate associated to the requested server in the Certificate Collector; if present the certificate will be used if not it will be automatically downloaded, properly modified and stored locally for future usage .
6) Packets from the victim are modified so that they are re-directed to the local acceptor socket; modifications are made on MAC addresses, IP addresses and TCP source ports (Port Address Translation "PAT" is used to handle multiple connections). The data captured is then sent again into the network using Winpcap but it is this time addressed to the local socket that will accept the Client-side connection.
7) The Server-side socket is created and connected to the real server requested by the victim.
8) OpenSSL libraries are used to manage encryption on both sockets using the fake certificate victim-side and the real certificate sever-side.
9) Packets sent by the Client-side socket are modified again to reach the victim's host.
10) Data coming from the server is decrypted, saved to session files, re-encrypted and sent to the victim host by mean of the Client-side socket.
11) Data coming from the client is decrypted, saved to session files, re-encrypted and sent to the server by mean of the Server-side socket.

Although it can be noticed from the fake certificate file used, this kind of attack is STEALTH from a client point of view because the victim thinks to be connected to the real server; try a "netstat -an" on the client to check yourself.

Once decrypted, traffic from the client is also sent to the HTTP sniffer filter for a further analysis on credentials. You can take a look

at the data saved in session files by APR-HTTPS here.

Prerequisites

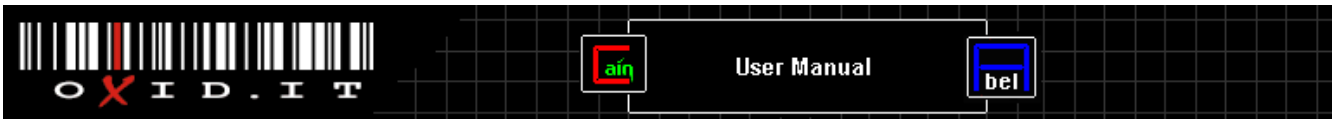
This feature needs APR to be enabled and a Man-in-the-Middle condition between the HTTPS server and the victim host.

Limitations

This feature does not work like a PROXY server; because of the usage of the Winpcap driver it cannot decrypt HTTPS sessions initiated from the local host.

Usage

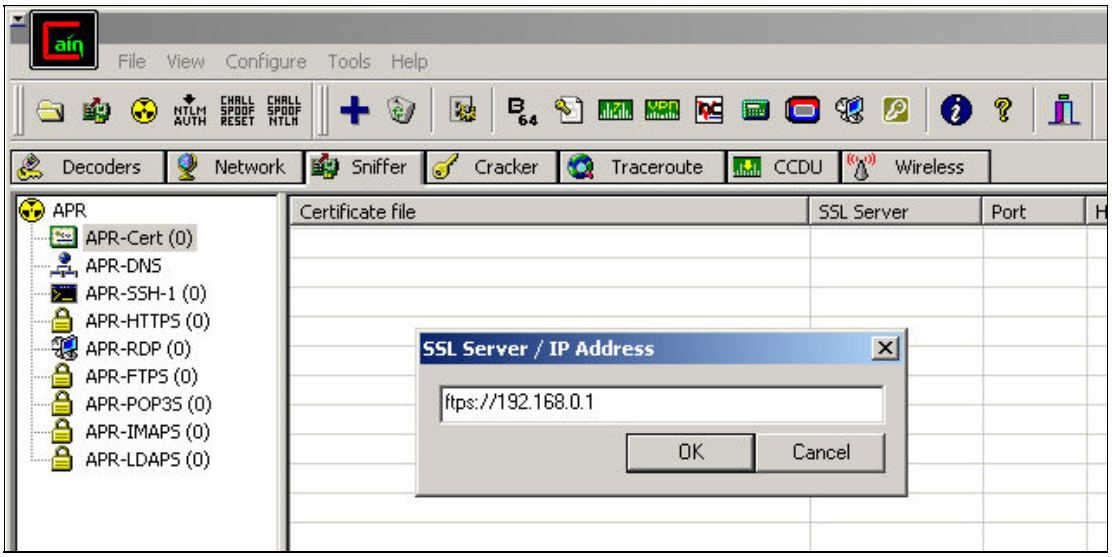
After you successfully set up APR and enabled the HTTPS sniffer filter, sessions are automatically saved in the HTTPS subdirectory and can be viewed using the relative function within the list pop up menu.



APR-FTPS

APR-FTPS enables the capture and the decryption of FTPS traffic between hosts. It works in conjunction with Cain's Certificate Collector to inject fake certificates into SSL sessions, previously hijacked by mean of APR. Using this trick it is possible to decrypt encrypted data before it arrives to the real destination performing a what so called Man-in-the-Middle attack.

Actually this feature support "implicit" FTPS protocol only which by default use TCP port 990. You can grab and prepare spoofed certificates in advance using Cain's Certificate Collector as show in the following picture.



Prerequisites

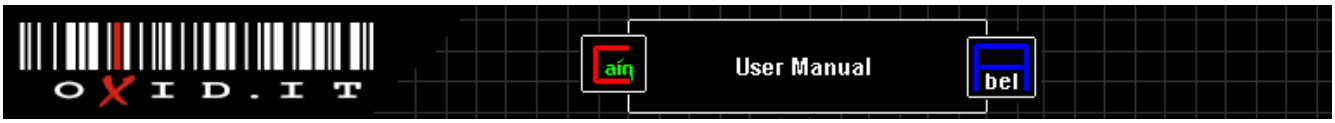
This feature needs APR to be enabled and a Man-in-the-Middle condition between the FTPS server and the victim host.
APR SSL spoofing features requires a direct TCP/IP connection between the attacker machine and the SSL enabled server. To handle spoofed communications Cain must be able to reach the remote SSL server without the use of proxy servers.

Limitations

This feature does not work like a PROXY server; because of the usage of the Winpcap driver it cannot decrypt FTPS sessions initiated from the local host.

Usage

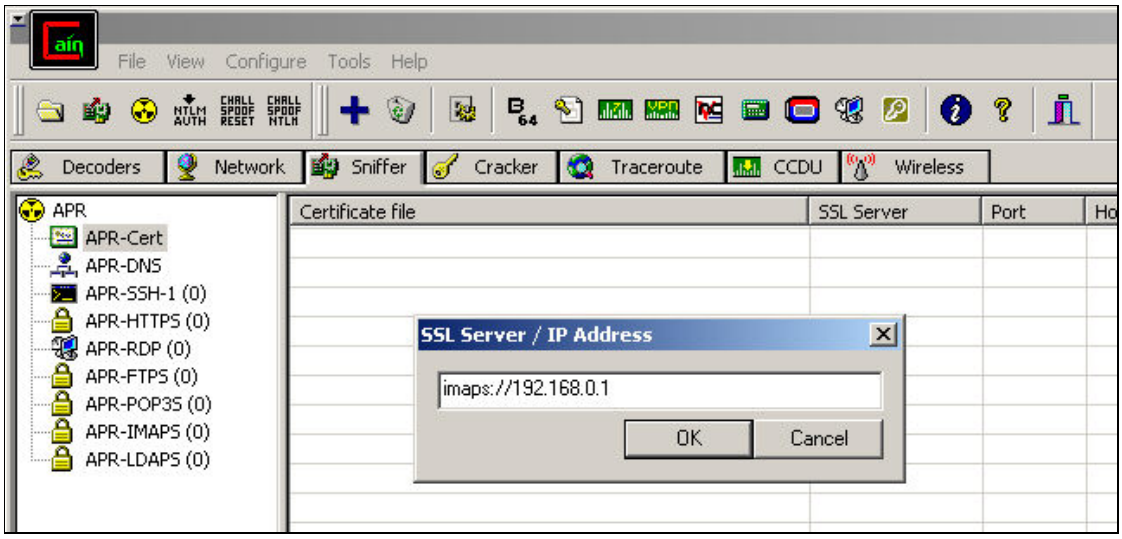
After you successfully set up APR enabling FTPS sniffer filter, sessions are automatically saved in the FTPS subdirectory and can be viewed using the relative function within the list pop up menu.



APR-IMAPS

APR-IMAPS enables the capture and the decryption of IMAPS traffic between hosts. It works in conjunction with Cain's Certificate Collector to inject fake certificates into SSL sessions, previously hijacked by mean of APR. Using this trick it is possible to decrypt encrypted data before it arrives to the real destination performing a what so called Man-in-the-Middle attack.

Actually this feature support "implicit" IMAPS protocol only which by default use TCP port 993. You can grab and prepare spoofed certificates in advance using Cain's Certificate Collector as show in the following picture.



Prerequisites

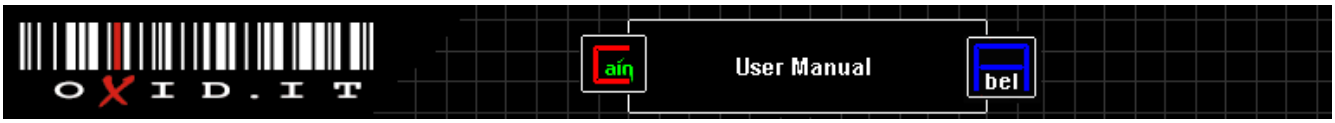
This feature needs APR to be enabled and a Man-in-the-Middle condition between the IMAPS server and the victim host.
APR SSL spoofing features requires a direct TCP/IP connection between the attacker machine and the SSL enabled server. To handle spoofed communications Cain must be able to reach the remote SSL server without the use of proxy servers.

Limitations

This feature does not work like a PROXY server; because of the usage of the Winpcap driver it cannot decrypt IMAPS sessions initiated from the local host.

Usage

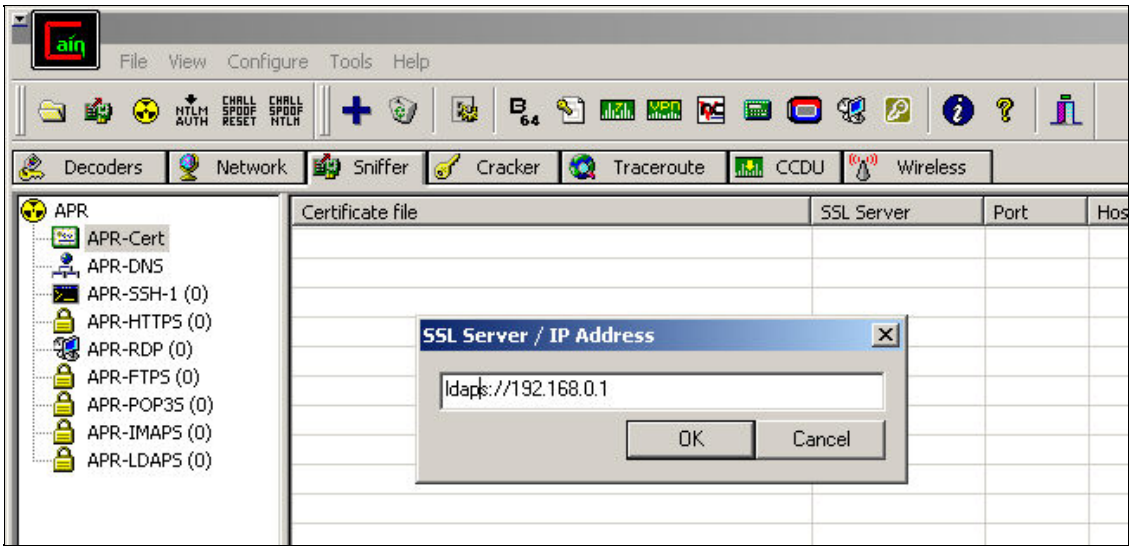
After you successfully set up APR enabling IMAPS sniffer filter, sessions are automatically saved in the IMAPS subdirectory and can be viewed using the relative function within the list pop up menu.



APR-LDAPS

APR-LDAPS enables the capture and the decryption of LDAPS traffic between hosts. It works in conjunction with Cain's Certificate Collector to inject fake certificates into SSL sessions, previously hijacked by mean of APR. Using this trick it is possible to decrypt encrypted data before it arrives to the real destination performing a what so called Man-in-the-Middle attack.

Actually this feature support "implicit" LDAPS protocol only which by default use TCP port 636. You can grab and prepare spoofed certificates in advance using Cain's Certificate Collector as show in the following picture.



Prerequisites

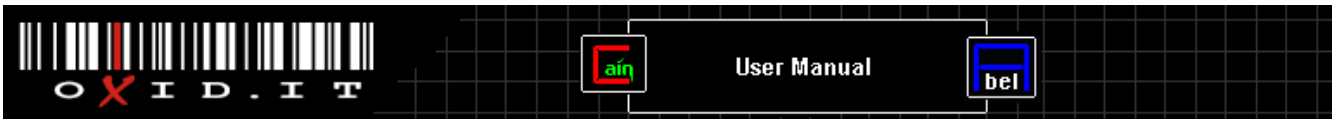
This feature needs APR to be enabled and a Man-in-the-Middle condition between the LDAPS server and the victim host.
APR SSL spoofing features requires a direct TCP/IP connection between the attacker machine and the SSL enabled server. To handle spoofed communications Cain must be able to reach the remote SSL server without the use of proxy servers.

Limitations

This feature does not work like a PROXY server; because of the usage of the Winpcap driver it cannot decrypt LDAPS sessions initiated from the local host.

Usage

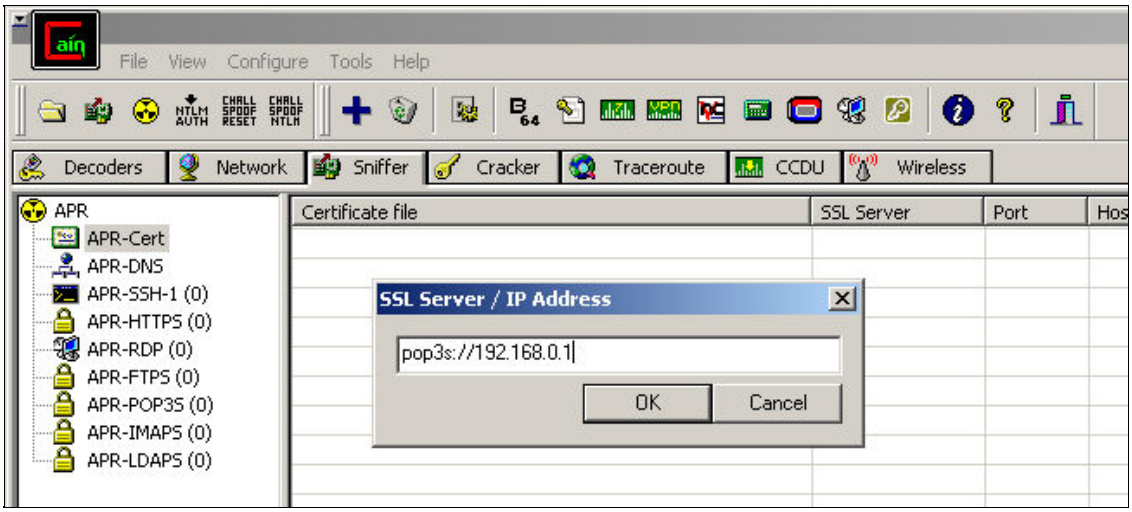
After you successfully set up APR enabling LDAPS sniffer filter, sessions are automatically saved in the LDAPS subdirectory and can be viewed using the relative function within the list pop up menu.



APR-POP3S

APR-POP3S enables the capture and the decryption of POP3S traffic between hosts. It works in conjunction with Cain's Certificate Collector to inject fake certificates into SSL sessions, previously hijacked by mean of APR. Using this trick it is possible to decrypt encrypted data before it arrives to the real destination performing a what so called Man-in-the-Middle attack.

Actually this feature support "implicit" POP3S protocol only which by default use TCP port 995. You can grab and prepare spoofed certificates in advance using Cain's Certificate Collector as show in the following picture.



Prerequisites

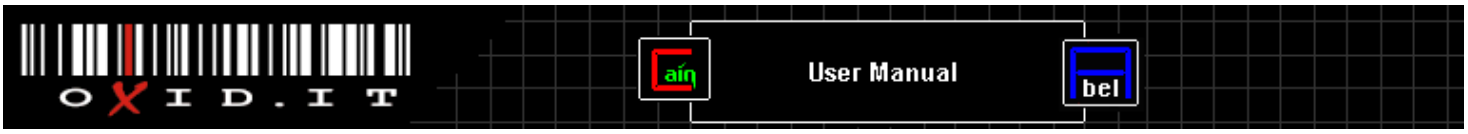
This feature needs APR to be enabled and a Man-in-the-Middle condition between the POP3S server and the victim host.
APR SSL spoofing features requires a direct TCP/IP connection between the attacker machine and the SSL enabled server. To handle spoofed communications Cain must be able to reach the remote SSL server without the use of proxy servers.

Limitations

This feature does not work like a PROXY server; because of the usage of the Winpcap driver it cannot decrypt POP3S sessions initiated from the local host.

Usage

After you successfully set up APR enabling POP3S sniffer filter, sessions are automatically saved in the POP3S subdirectory and can be viewed using the relative function within the list pop up menu.



APR-RDP

APR-RDP enables the capture and the decryption of Remote Desktop Protocol (RDP) traffic between hosts. RDP is the protocol used to connect to Windows Terminal Services of a remote computer.

Microsoft's Windows Terminal Services (built into Windows 2000 Server and Windows Server 2003) and Windows XP's Remote Desktop, provide an easy, convenient way for administrators to implement thin computing within an organization or for users to connect to their XP desktops from a remote computer and run applications or access files.
A Windows 2000 terminal server can be installed in one of two modes: administrative or application server. In administrative mode, only users with administrative accounts can access the terminal server .... this is why these sessions are so interesting.

By default, data that travels between the terminal server and the terminal services client is protected by encryption. The protocol uses the RC4 symmetric encryption algorithm
at one of the following three levels:

- **High**: encrypts both the data sent from client to server and the data sent from server to client using a 128-bit key.
- **Medium**: encrypts both the data sent from client to server and the data sent from server to client using a 56-bit key if the client is a Windows 2000 or above client, or a 40-bit key if the client is an earlier version.
- **Low**: encrypts only the data sent from client to server, using either a 56-bit or 40-bit key, depending on the client version.

RC4 encryption keys are generated after an initial key exchange in which RSA asymmetric encryption is used.

In April 2003 Erik Forsberg released a security advisory to the public (http://www.securityfocus.com/archive/1/317244) explaining that:

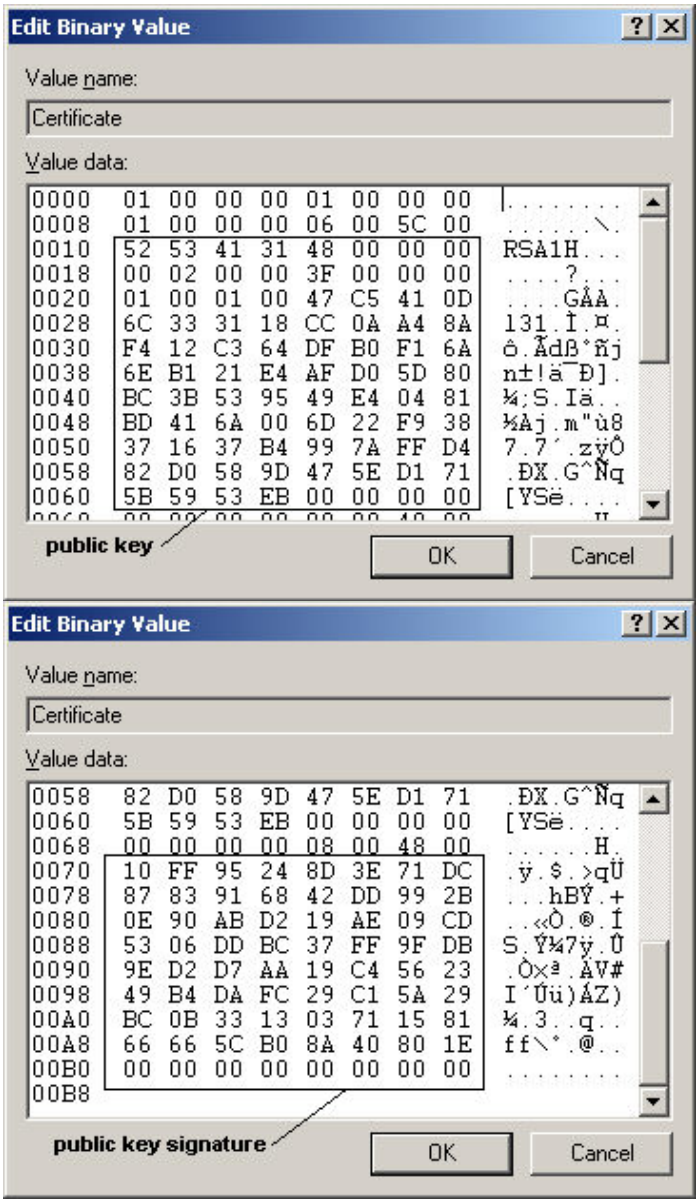
"*... During extensive investigation of the Remote Desktop Protocol (RDP), the protocol used to connect to Windows Terminal Services, we have found that although the information sent over the network is encrypted, there is no verification of the identity of the server when setting up the encryption keys for the session. This means RDP is vulnerable to Man In The Middle attacks (from here on referred to as MITM attacks). The attack works as follows:*
1) The client connects to the server, however by some method (DNS spoofing, arp poisioning, etc.) we've fooled it to connect to the MITM instead. The MITM sends the request further to the server.
2) The server sends it's public key and a random salt, in cleartext, again through the MITM. The MITM sends the packet further to the client, but exchanges the public key to another one for which it knows the private part.
3) The client sends a random salt, encrypted with the server public key, to the MITM.
4) The MITM deencrypts the clients random salt with it's private key, encrypts it with the real servers public key and sends it to the server.
5) The MITM now know both the server and the client salt, which is enough information to construct the session keys used for further packets sent between the client and the server. All information sent between the parts can now be read in cleartext.

The vulnerability occurs because the clients by no means try to verify the public key of the server, sent in step 2 above. In other protocols, such as the Secure Shell protocol, most client implementations solve this for example by letting the user answer a question whether a specific serverkey fingerprint is valid. *...*"

Microsoft confirmed the above problem and fixed the new versions of Remote Desktop Clients. Recent clients (mstsc.exe), including the one of version XPSP2 5.1.2600.2180, now check the Terminal Server identity verifying its public key. They solved the problem ? No, **man-in-the-middle attacks are still possible and can be really invisible for users**.

During the initial key-exchange phase, the terminal server sends to the client a server certificate created at the start up of Terminal Server services. This certificate is stored in the registry of the server under the following key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Parameters\Certificate

It contains an RSA public key and its digital signature as illustrated below:



The public key modulus (n) is the same as the one present in the RSA2 key stored in the LSA Secret "L$HYDRAENCKEY" (you can use the Cain's LSA Secret Dumper to check it) of the server; the signature is the information used by the client to verify the server identity.
From a man-in-the-middle attacker's point of view, the public key signature must be modified on the fly to trick the client into verifying the new Mitm public key that will be replaced into the network packet directed to the client. But … what is used to produce this signature ?

Well, a digital signature is noting more nothing less than a hash of something (in this case a server public key) encrypted using a private key and an asymmetric encryption algorithm. This is exactly what is done by the terminal server. At the client-side, this signature is decrypted using a public key and the result is compared with a new hash of the received server public key calculated by the client; if the two hashes match the identity of the server is proven.

Microsoft use another RSA private key to sign the Terminal Server public key and this private key is public ! It could sound strange but this is only the truth, the private key used for the signature creation is hard-coded into mstlsapi.dll and it is dynamically created, used and de-allocated into a subroutine of the "TLSInit" API. Every Windows user has this file ... is this a new kind of public-private key (PPK) ?!?

The Microsoft Windows Terminal Server PPK follows:

public exponent: e
0x5B,0x7B,0x88,0xC0

public modulus: n
0x3D,0x3A,0x5E,0xBD,0x72,0x43,0x3E,0xC9,0x4D,0xBB,0xC1,0x1E,0x4A,0xBA,0x5F,0xCB,
0x3E,0x88,0x20,0x87,0xEF,0xF5,0xC1,0xE2,0xD7,0xB7,0x6B,0x9A,0xF2,0x52,0x45,0x95,
0xCE,0x63,0x65,0x6B,0x58,0x3A,0xFE,0xEF,0x7C,0xE7,0xBF,0xFE,0x3D,0xF6,0x5C,0x7D,
0x6C,0x5E,0x06,0x09,0x1A,0xF5,0x61,0xBB,0x20,0x93,0x09,0x5F,0x05,0x6D,0xEA,0x87,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00

private exponent: d
0x87,0xA7,0x19,0x32,0xDA,0x11,0x87,0x55,0x58,0x00,0x16,0x16,0x25,0x65,0x68,0xF8,
0x24,0x3E,0xE6,0xFA,0xE9,0x67,0x49,0x94,0xCF,0x92,0xCC,0x33,0x99,0xE8,0x08,0x60,
0x17,0x9A,0x12,0x9F,0x24,0xDD,0xB1,0x24,0x99,0xC7,0x3A,0xB8,0x0A,0x7B,0x0D,0xDD,
0x35,0x07,0x79,0x17,0x0B,0x51,0x9B,0xB3,0xC7,0x10,0x01,0x13,0xE7,0x3F,0xF3,0x5F,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00

secret prime factor: p
0x3F,0xBD,0x29,0x20,0x57,0xD2,0x3B,0xF1,0x07,0xFA,0xDF,0xC1,0x16,0x31,0xE4,0x95,
0xEA,0xC1,0x2A,0x46,0x2B,0xAD,0x88,0x57,0x55,0xF0,0x57,0x58,0xC6,0x6F,0x95,0xEB,
0x00,0x00,0x00,0x00

secret prime factor: q
0x83,0xDD,0x9D,0xD0,0x03,0xB1,0x5A,0x9B,0x9E,0xB4,0x63,0x02,0x43,0x3E,0xDF,0xB0,
0x52,0x83,0x5F,0x6A,0x03,0xE7,0xD6,0x78,0x45,0x83,0x6A,0x5B,0xC4,0xCB,0xB1,0x93,
0x00,0x00,0x00,0x00

d mod (p-1): dmp1
0x65,0x9D,0x43,0xE8,0x48,0x17,0xCD,0x29,0x7E,0xB9,0x26,0x5C,0x79,0x66,0x58,0x61,
0x72,0x86,0x6A,0xA3,0x63,0xAD,0x63,0xB8,0xE1,0x80,0x4C,0x0F,0x36,0x7D,0xD9,0xA6,
0x00,0x00,0x00,0x00

d mod (q-1): dmq1
0x75,0x3F,0xEF,0x5A,0x01,0x5F,0xF6,0x0E,0xD7,0xCD,0x59,0x1C,0xC6,0xEC,0xDE,0xF3,
0x5A,0x03,0x09,0xFF,0xF5,0x23,0xCC,0x90,0x27,0x1D,0xAA,0x29,0x60,0xDE,0x05,0x6E,
0x00,0x00,0x00,0x00

q^-1 mod p: iqmp
0xC0,0x17,0x0E,0x57,0xF8,0x9E,0xD9,0x5C,0xF5,0xB9,0x3A,0xFC,0x0E,0xE2,0x33,0x27,
0x59,0x1D,0xD0,0x97,0x4A,0xB1,0xB1,0x1F,0xC3,0x37,0xD1,0xD6,0xE6,0x9B,0x35,0xAB,
0x00,0x00,0x00,0x00

The knowledge of the PPK key lets the attacker calculate a valid signature for the mitm public key generated on the fly during the mitm attack; the client will verify the mitm signature correctly and it will accept the session without informing the users that the server key is changed from the usual one.
The signature is calculated encrypting, with the private part of the PPK key, the MD5 hash of the server public key for a total of 108 bytes hashed.

How it works

0) The network packet from the server is hijacked and captured by mean of APR (ARP Poison Routing).
1) The server random and the real server public key are extracted from the packet and stored for future usage.
2) The server public key is replaced in the network packet with a new one generated by Cain (the mitm machine) during the key exchange phase.
3) The MD5 hash of the new mitm public key is calculated.
4) The hash is signed by Cain (encrypted using the private key) using the super secret Microsoft PPK illustrated above.
5) The mitm sign is replaced into the network packet.
6) The packet is routed by APR to the client.
7) The network packet from the client is hijacked and captured by mean of APR (ARP Poison Routing).
8) The client encrypted random is decrypted using the mitm private key.
9) The client random is encrypted using the real server public key and replaced into the network packet for the server.
10) The packet is routed by APR to the server.
11) RC4 symmetric encryption keys are calculated.
12) The key entropy is reduced accordingly with the encryption level used in the session.

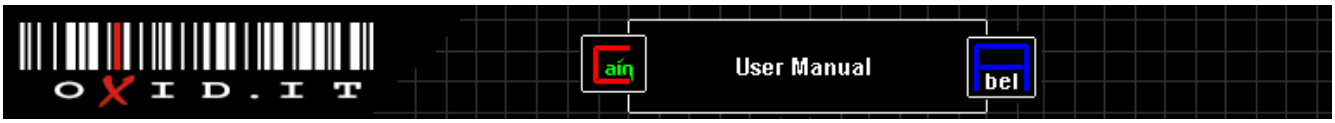
13) Packets are decrypted and saved locally to text files.

Authentication

Cain also try to recognize the keyboard activity at the client-side. This provide some kind of password interception.

Prerequisites

This feature needs APR to be enabled and a Man-in-the-Middle condition between the Terminal Server and the victim host.



APR-SSH-1

Well documented in the RFC "**The SSH (Secure Shell) Remote Login Protocol**", SSH provides strong authentication and secure communications over insecure networks.
APR-SSH-1 uses the Man-in-the-Middle condition imposed by Cain's APR to capture and decrypt SSH (Secure Shell) traffic between hosts.

How it works

The SSH protocol is composed by several phases that can be summarized as follow:

- Connection

The client connects to the SSH port (usually TCP port 22) of the server.

- Identification phase

The server sends to the client an identification string of the form "SSH-<protocolmajor>.<protocolminor>-<version>"; the client parses the server's string, and sends a corresponding string with its own information in response. Here APR-SSH-1 automatically replaces the version specified by the server in the first packet in order to downgrade the communication to SSH protocol v1.51.

- Negotiation phase

During this phase the server sends its asymmetric encryption keys and other parameters to the client. Cain collects server's keys and replace them with new ones generated locally so that the client will use the Cain's keys instead of the server's ones.

- Session setup phase

The client selects the symmetric cipher to use, and sends the encrypted session key to the server. At this point the session key has been encrypted by the client using the Cain's keys, not the server keys; for this reason the program can now decrypt and store the session key before send it back to the server. The session key is really important because it is used to decrypt all packets that follows.

- Encrypted phase

The server and the client start the secure communication using the specified symmetric cipher and the session key. Now Cain can use the session key to decrypt encrypted traffic capture on the network.

APR-SSH-1 works in FULL-DUPLEX mode processing both client and server SSH-1 traffic. Because of the use of APR (Arp Poison Routing), the attacker's IP and MAC addresses can be totally spoofed and never sent across the network. The sniffer supports tree symmetric encryption algorithms: DES, 3DES, Blowfish.

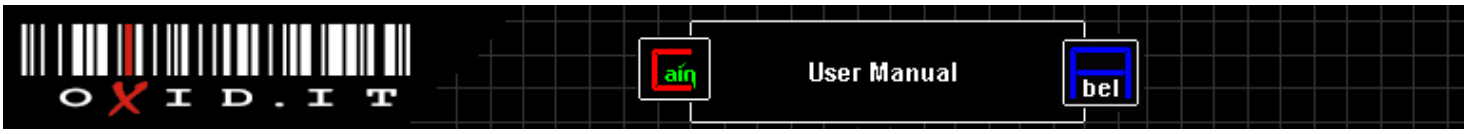
Prerequisites

This feature needs APR to be enabled and a Man-in-the-Middle condition between the SSH server and the victim host.

Limitations

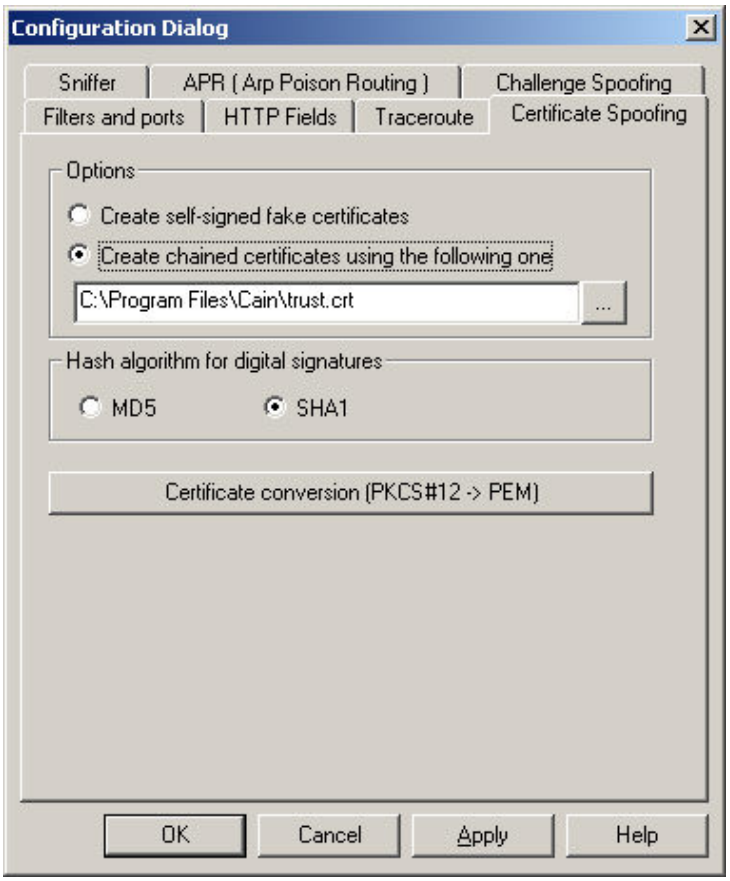
Zlib compression is not supported in this version. Because of the usage of the Winpcap driver it cannot decrypt SSH1 sessions initiated from the local host.

An example of the output file produced by the sniffer from the capture of an SSH-1 session to a Cisco PIX firewall in my test environment is available here.

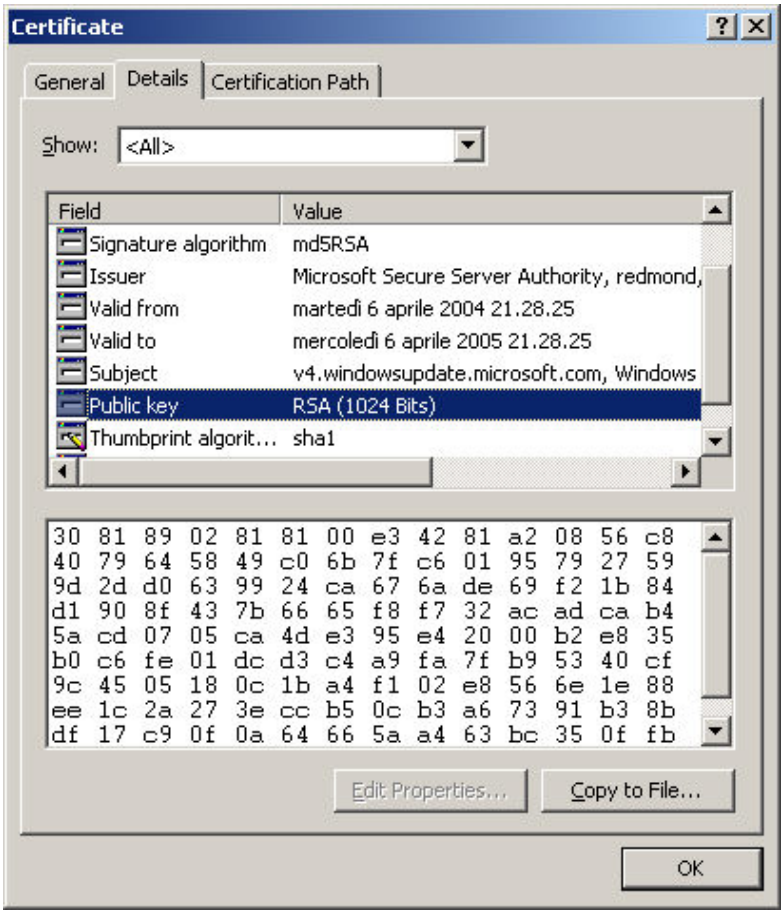


Certificates Collector

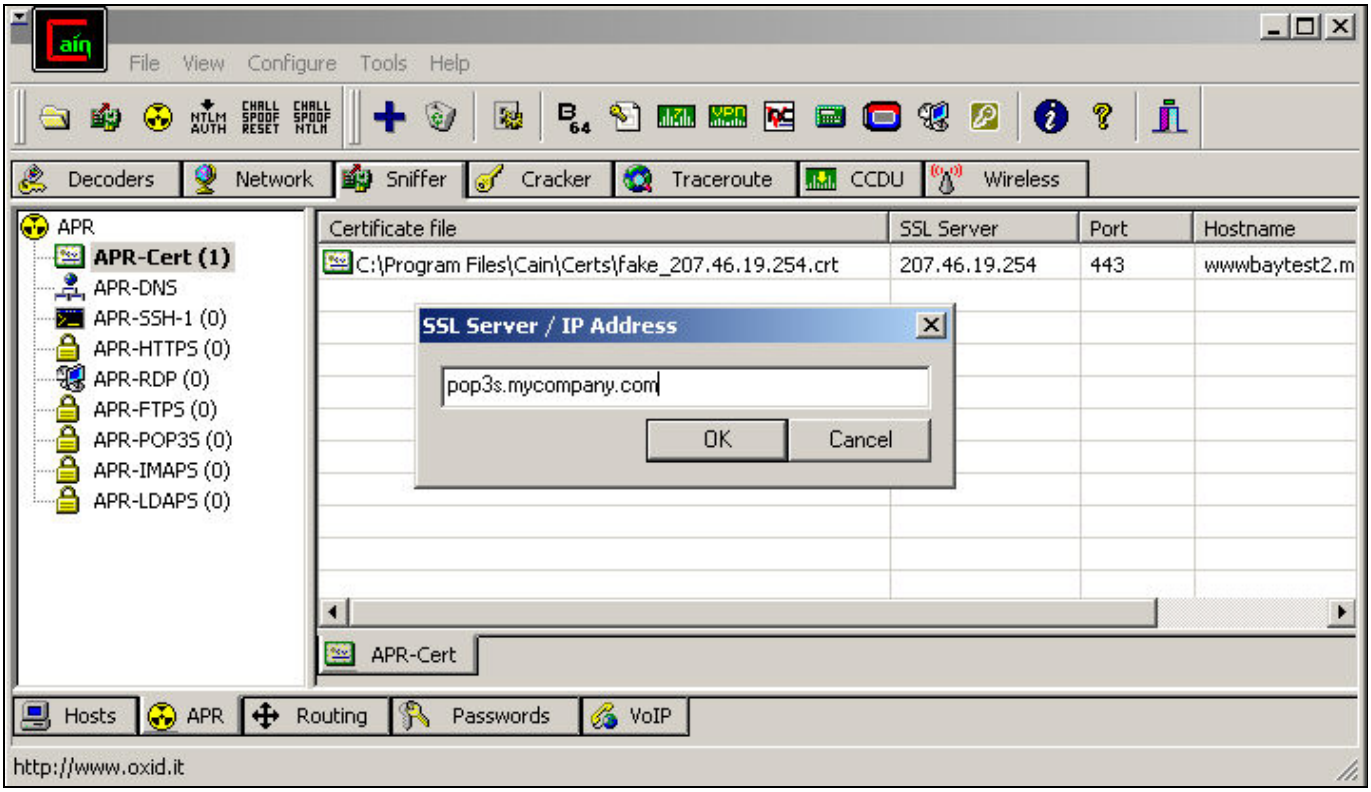
Cain's Certificates Collector grabs server certificates from SSL enabled sites and prepares them for APR-*S filters. The feature is automatically used by the sniffer but you can also manually create a list of pre-calculated fake certificate files. Why fake ? because the program will replace asymmetric encryption keys in these files with new ones generated locally. In this way the APR-*S filters will be able to encrypt/decrypt SSL traffic in a Man-in-the-Middle condition between victim APR's hosts.
You can choose to create/use self-signed fake certificates or to sign them using a real root certificate of your choice; in the last case the whole certificate chain will be injected to victims clients during MitM attacks.



When using self-signed fake certificates the client's browser is supposed to pop up a dialog to notify users that the SSL-enabled server certificate is coming from an untrusted certification authority; anyway since all other parameters within the certificate remain the same as the real ones a lot of users simply does not care about this warning.

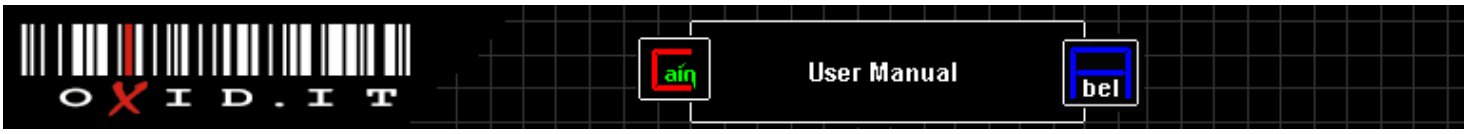


Fake certificates are stored in the "Certs" subdirectory of the program's installation path and the list of those currently available to APR-*S filters is maintained in the file CERT.LST in the program's directory. You can manually modify this list file to instruct Cain's APR-*S filters to inject the certificate of your choice into connections from APR's victims computers to a given SSL server address.



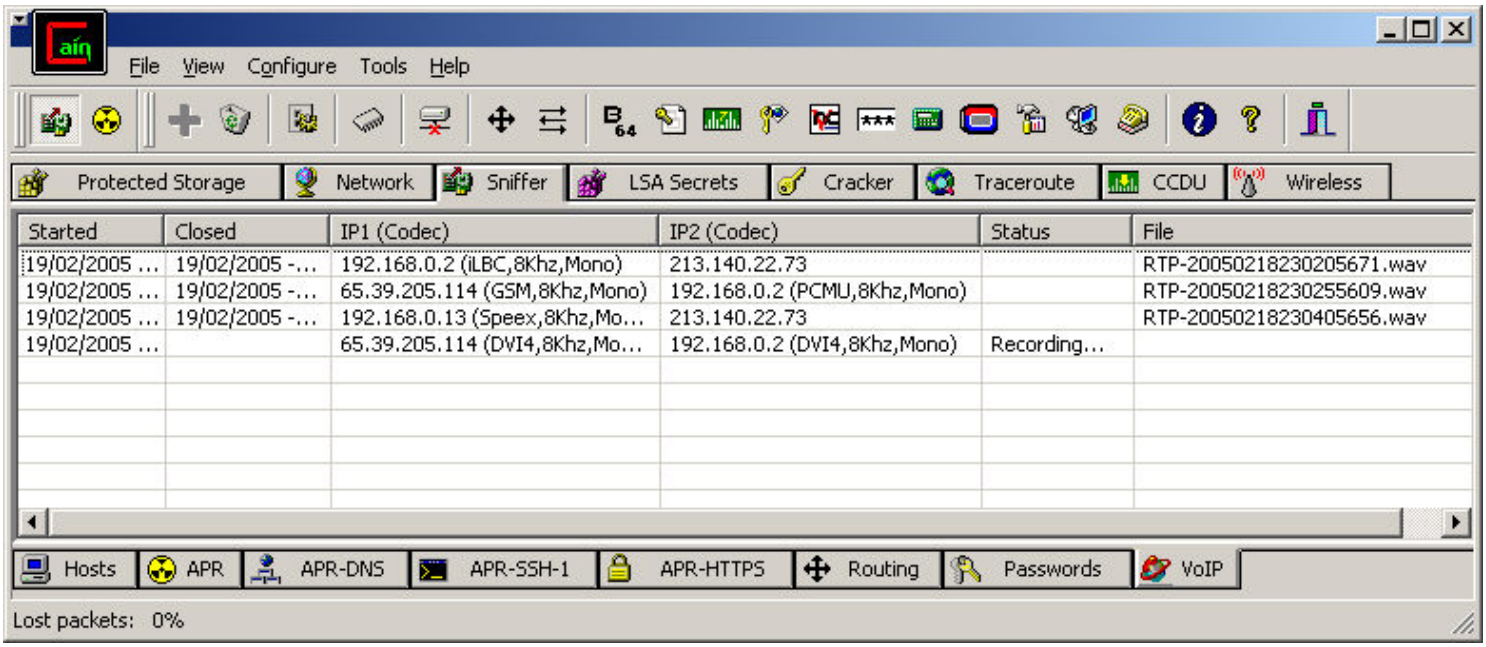
Usage

The feature is used automatically by the APR-*S sniffer filters. You can use the + button on the toolbar to manually grab and prepare a list of fake certificates; non standard ports can be specified using the syntax "hostname:port" or "ip address:port".



VoIP

The VoIP (Voice over IP) sniffer captures conversations from the network and records them to your hard disk. If seen by the sniffer, voice data is captured in each direction (caller<->responder) and then saved accordingly as mono or stereo WAV files. Although not required, if used with APR, this feature enables to silently intercept VoIP communications between victim hosts.



A lot of VoIP solutions both hardware and software are available as of today; some applications are:

- Microsoft Messenger (http://www.microsoft.com)
- X-Lite softphone (http://www.xten.com)
- Pulver communicator (http://www.freeworlddialup.com)
- KPhone (http://www.wirlab.net/kphone)
- Gnomemeeting (http://www.gnomemeeting.org)
- eStara softphone (http://www.estara.com)
- Advanced Dialer (http://www.advanceddialer.com)
- Pingtel SIP Softphone (http://www.pingtel.com)
- SIPPS (http://www.sippstar.com)
- OpenH323 (http://www.openh323.org)
- Asterisk (http://www.asterisk.org)
- PhoneGaim (http://phonegaim.com)
- SJphone (http://www.sjlabs.com)
- .....

VoIP applications make use of signaling protocols, like H323 and SIP, for creating, modifying and terminating sessions with conversation's participants.
Voice data streams are usually transmitted on the Internet by mean of the RTP (Real-Time Transport Protocol) protocol. RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. Speech data is sent in compressed forms to preserve bandwidth; codecs (Compressor/decompressor) are used to convert the data streams at the transmitter/receiver side.
The RFC-3551 describes how audio and video data may be carried within RTP and it also defines a set of standard encodings, and their names, when used within RTP.

How it works

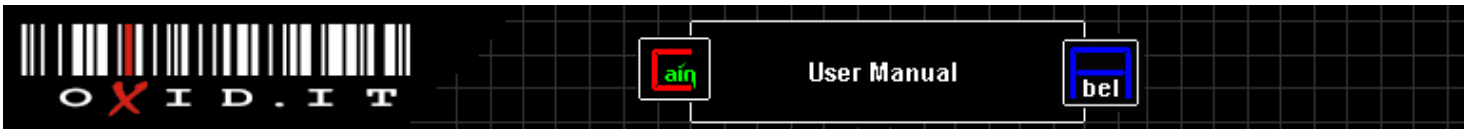
The sniffer extracts RTP session parameters like RTP ports, caller/responder IP addresses and dynamic codec types from SIP or MGCP session preceding the data flow on RTP. Then it captures and decodes RTP audio streams encoded with the following codecs: G711 uLaw, G771 aLaw, ADPCM, DVI4, LPC, GSM610, Microsoft GSM, L16, G729, Speex, iLBC, G722.1, G723.1, G726-16, G726-24, G726-32, G726-40, LPC-10, SIREN, LRWB-16khz. Once decoded the audio is saved into mono or stereo WAV files on your hard disk.

Note

The sniffer can decode and save VoIP conversations for supported codecs only.

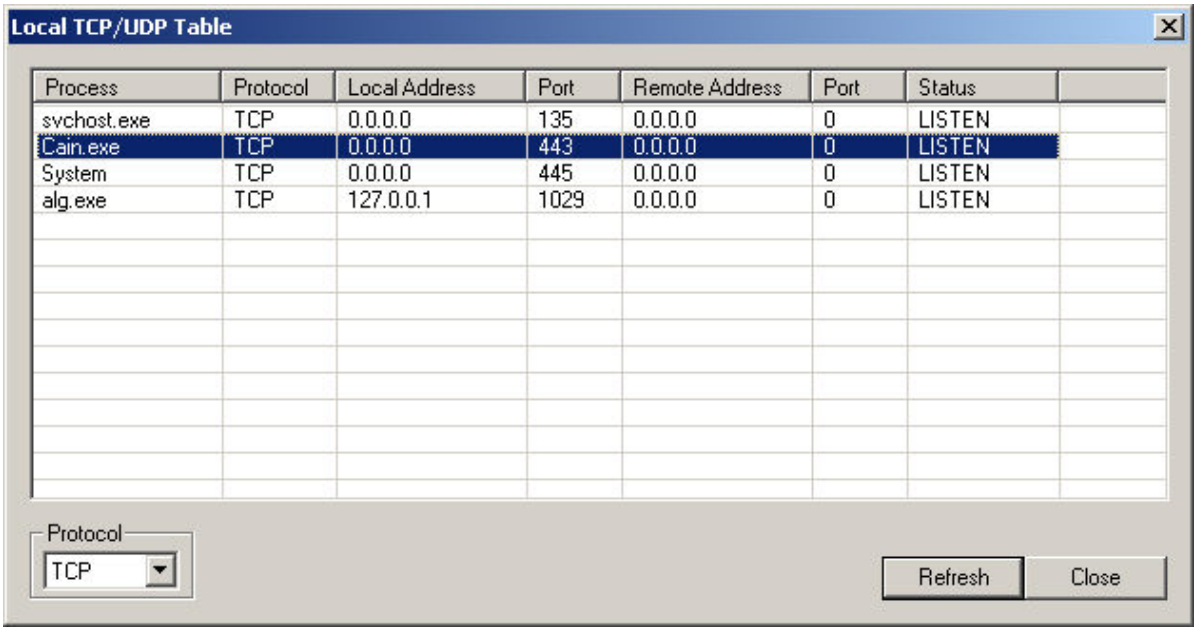
Usage

Enable the SIP/RTP sniffer filter is the sniffer's configuration dialog.



TCP/UDP Table Viewer

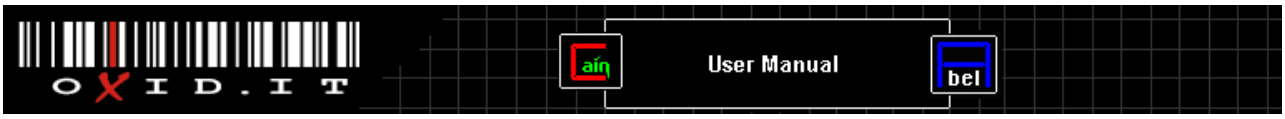
Cain's TCP/UDP Table viewer covers the same functionality offered by the Windows tool "netstat.exe".



The dialog shows all connection's parameters as protocol type, local address, port number, remote address and connection's status (TCP only). On Windows XP, the filename of the process associated to TCP and UDP ports is also displayed.

Note

When executed, Cain binds a TCP socket on port 443 on the local machine. This is needed by the APR-HTTPS feature present in the program; Cain does not open any other local port in listening mode. For more information please refer to this page.

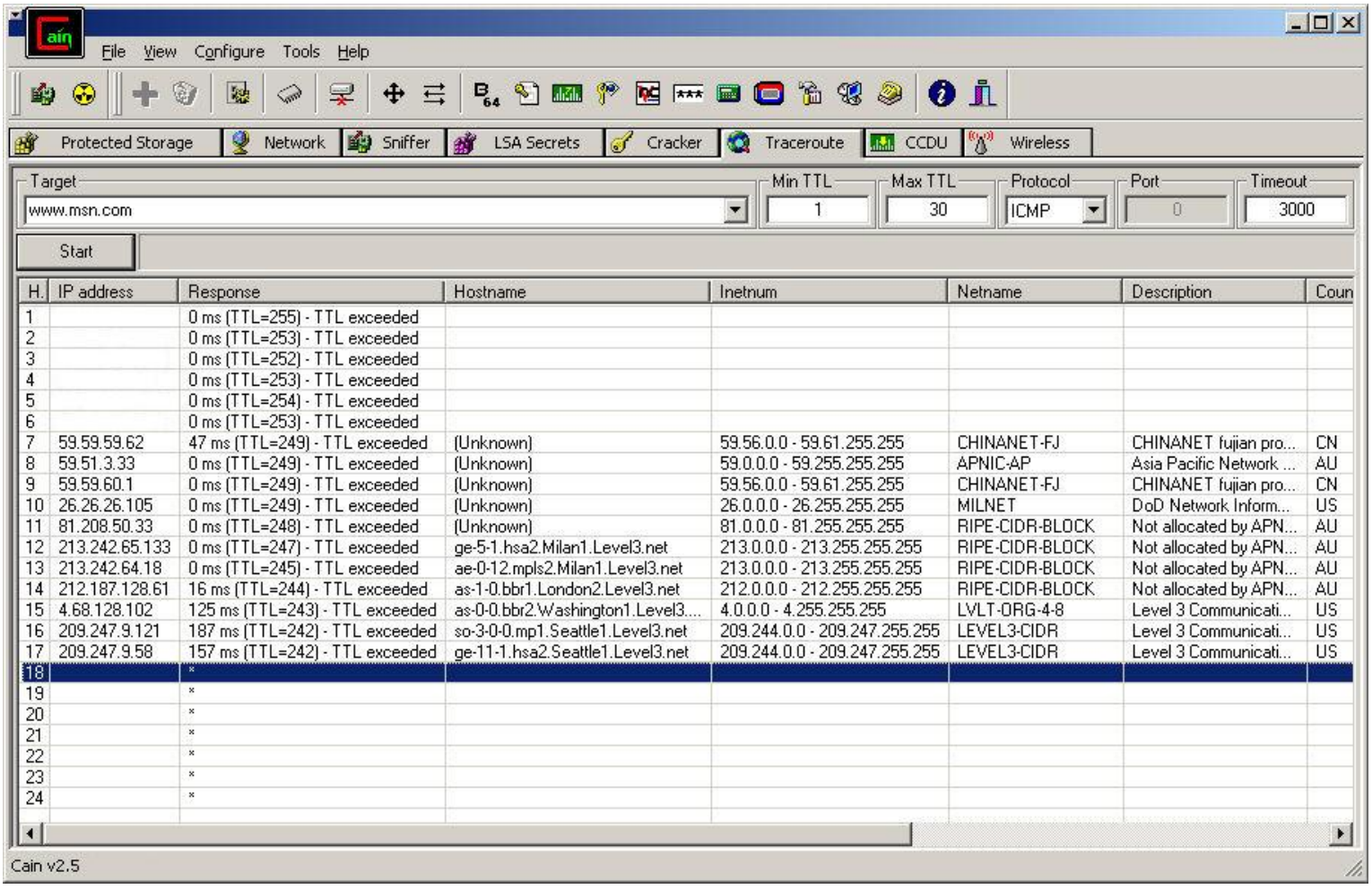


Traceroute

Cain's traceroute is an improved version of the Windows tool "tracert.exe".

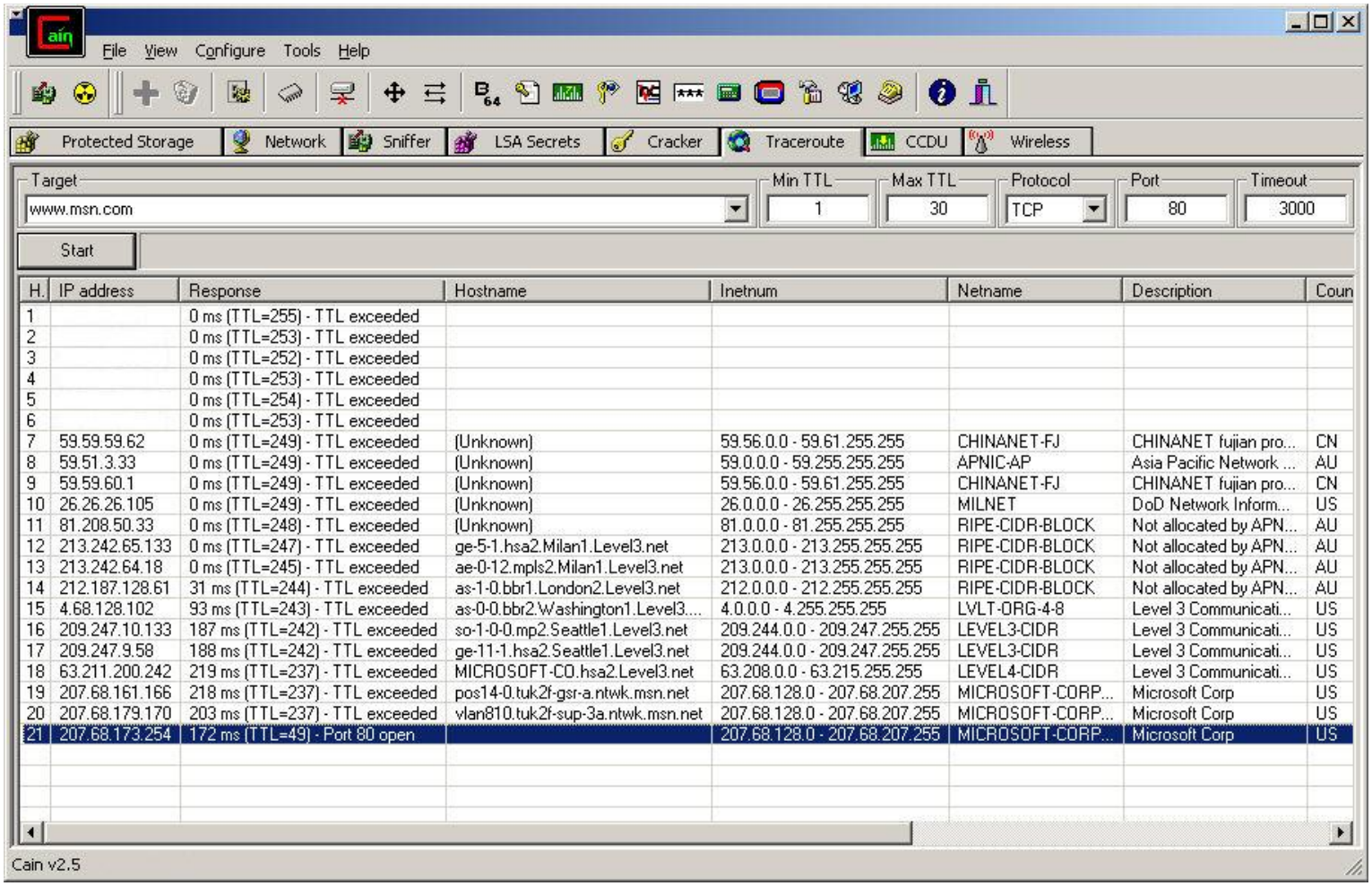
The widespread usage of perimeter defences on the modern Internet makes sometimes impossible to reach the desired destination using the above utility; firewalls can drop ICMP packets without sending back ICMP responses, for this reason the entire path to the target host could not be completely traced. UDP or TCP protocol can be used to bypass common firewall restrictions so Cain's traceroute supports all of them.

Consider for example the following ICMP trace to **www.msn.com**:



H.	IP address	Response	Hostname	Inetnum	Netname	Description	Coun
1		0 ms (TTL=255) - TTL exceeded					
2		0 ms (TTL=253) - TTL exceeded					
3		0 ms (TTL=252) - TTL exceeded					
4		0 ms (TTL=253) - TTL exceeded					
5		0 ms (TTL=254) - TTL exceeded					
6		0 ms (TTL=253) - TTL exceeded					
7	59.59.59.62	47 ms (TTL=249) - TTL exceeded	(Unknown)	59.56.0.0 - 59.61.255.255	CHINANET-FJ	CHINANET fujian pro...	CN
8	59.51.3.33	0 ms (TTL=249) - TTL exceeded	(Unknown)	59.0.0.0 - 59.255.255.255	APNIC-AP	Asia Pacific Network ...	AU
9	59.59.60.1	0 ms (TTL=249) - TTL exceeded	(Unknown)	59.56.0.0 - 59.61.255.255	CHINANET-FJ	CHINANET fujian pro...	CN
10	26.26.26.105	0 ms (TTL=249) - TTL exceeded	(Unknown)	26.0.0.0 - 26.255.255.255	MILNET	DoD Network Inform...	US
11	81.208.50.33	0 ms (TTL=248) - TTL exceeded	(Unknown)	81.0.0.0 - 81.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
12	213.242.65.133	0 ms (TTL=247) - TTL exceeded	ge-5-1.hsa2.Milan1.Level3.net	213.0.0.0 - 213.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
13	213.242.64.18	0 ms (TTL=245) - TTL exceeded	ae-0-12.mpls2.Milan1.Level3.net	213.0.0.0 - 213.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
14	212.187.128.61	16 ms (TTL=244) - TTL exceeded	as-1-0.bbr1.London2.Level3.net	212.0.0.0 - 212.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
15	4.68.128.102	125 ms (TTL=243) - TTL exceeded	as-0-0.bbr2.Washington1.Level3....	4.0.0.0 - 4.255.255.255	LVLT-ORG-4-8	Level 3 Communicati...	US
16	209.247.9.121	187 ms (TTL=242) - TTL exceeded	so-3-0-0.mp1.Seattle1.Level3.net	209.244.0.0 - 209.247.255.255	LEVEL3-CIDR	Level 3 Communicati...	US
17	209.247.9.58	157 ms (TTL=242) - TTL exceeded	ge-11-1.hsa2.Seattle1.Level3.net	209.244.0.0 - 209.247.255.255	LEVEL3-CIDR	Level 3 Communicati...	US
18		*					
19		*					
20		*					
21		*					
22		*					
23		*					
24		*					

The ICMP traceroute stops at hop 18; probably there is something over there that drops ICMP packets. The same trace but this time using TCP packets will cross that firewall entering in the Microsoft Network.



H.	IP address	Response	Hostname	Inetnum	Netname	Description	Coun
1		0 ms (TTL=255) - TTL exceeded					
2		0 ms (TTL=253) - TTL exceeded					
3		0 ms (TTL=252) - TTL exceeded					
4		0 ms (TTL=253) - TTL exceeded					
5		0 ms (TTL=254) - TTL exceeded					
6		0 ms (TTL=253) - TTL exceeded					
7	59.59.59.62	0 ms (TTL=249) - TTL exceeded	(Unknown)	59.56.0.0 - 59.61.255.255	CHINANET-FJ	CHINANET fujian pro...	CN
8	59.51.3.33	0 ms (TTL=249) - TTL exceeded	(Unknown)	59.0.0.0 - 59.255.255.255	APNIC-AP	Asia Pacific Network ...	AU
9	59.59.60.1	0 ms (TTL=249) - TTL exceeded	(Unknown)	59.56.0.0 - 59.61.255.255	CHINANET-FJ	CHINANET fujian pro...	CN
10	26.26.26.105	0 ms (TTL=249) - TTL exceeded	(Unknown)	26.0.0.0 - 26.255.255.255	MILNET	DoD Network Inform...	US
11	81.208.50.33	0 ms (TTL=248) - TTL exceeded	(Unknown)	81.0.0.0 - 81.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
12	213.242.65.133	0 ms (TTL=247) - TTL exceeded	ge-5-1.hsa2.Milan1.Level3.net	213.0.0.0 - 213.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
13	213.242.64.18	0 ms (TTL=245) - TTL exceeded	ae-0-12.mpls2.Milan1.Level3.net	213.0.0.0 - 213.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
14	212.187.128.61	31 ms (TTL=244) - TTL exceeded	as-1-0.bbr1.London2.Level3.net	212.0.0.0 - 212.255.255.255	RIPE-CIDR-BLOCK	Not allocated by APN...	AU
15	4.68.128.102	93 ms (TTL=243) - TTL exceeded	as-0-0.bbr2.Washington1.Level3....	4.0.0.0 - 4.255.255.255	LVLT-ORG-4-8	Level 3 Communicati...	US
16	209.247.10.133	187 ms (TTL=242) - TTL exceeded	so-1-0-0.mp2.Seattle1.Level3.net	209.244.0.0 - 209.247.255.255	LEVEL3-CIDR	Level 3 Communicati...	US
17	209.247.9.58	188 ms (TTL=242) - TTL exceeded	ge-11-1.hsa2.Seattle1.Level3.net	209.244.0.0 - 209.247.255.255	LEVEL3-CIDR	Level 3 Communicati...	US
18	63.211.200.242	219 ms (TTL=237) - TTL exceeded	MICROSOFT-CO.hsa2.Level3.net	63.208.0.0 - 63.215.255.255	LEVEL4-CIDR	Level 3 Communicati...	US
19	207.68.161.166	218 ms (TTL=237) - TTL exceeded	pos14-0.tuk2f-gsr-a.ntwk.msn.net	207.68.128.0 - 207.68.207.255	MICROSOFT-CORP...	Microsoft Corp	US
20	207.68.179.170	203 ms (TTL=237) - TTL exceeded	vlan810.tuk2f-sup-3a.ntwk.msn.net	207.68.128.0 - 207.68.207.255	MICROSOFT-CORP...	Microsoft Corp	US
21	207.68.173.254	172 ms (TTL=49) - Port 80 open		207.68.128.0 - 207.68.207.255	MICROSOFT-CORP...	Microsoft Corp	US

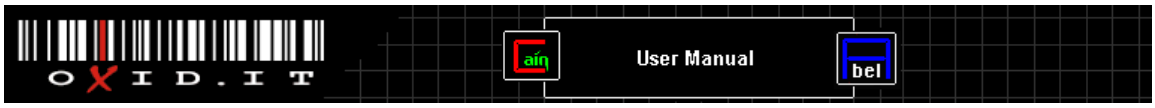
As you can see the TCP traceroute reached the destination host (www.msn.com) discovering some routers inside the Microsoft Corporation.

Requirements

Windows firewall could prevent this feature to work as expected.

Usage

Choose the protocol type, select the target and press start.

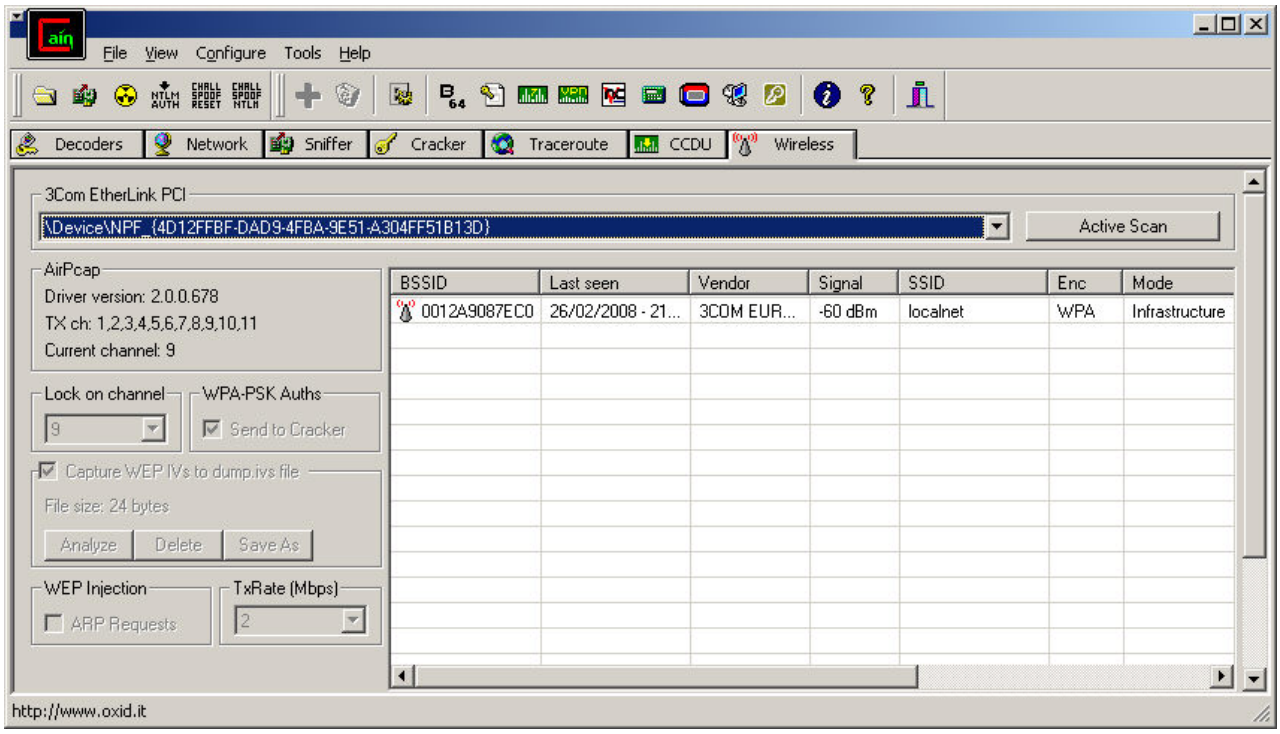


Wireless Scanner

Cain's Wireless Scanner detects Wireless Local Area Networks (WLANs) using 802.11x.

Active Scanner

Unlike other wireless applications it does not use the Windows NDIS User Mode I/O Protocol (NDISUIO) but the Winpcap Packet Driver to control the wireless network card. Access points and ah-hoc networks are enumerated using 802.11 OIDs from Windows DDK at intervals of five seconds and WLANs parameters (MAC address, SSID, Vendor, WEP Encryption, Channels....) are displayed in the scanner list.



How Active Scanner works

The active scanner opens the wireless network adapter using the Winpcap protocol driver then it uses the "PacketRequest" function of the same driver to communicate with the wireless network card. This API can be used from the Windows User Mode to perform a query/set operation on an internal variable of the network card driver.

BOOL PacketRequest (LPADAPTER AdapterObject, BOOL Set, PPACKET_OID_DATA OidData);

...from Winpcap documentation
not all the network adapters implement all the query/set functions. There is a set of mandatory OID functions that is granted to be present on all the adapters, and a set of facultative functions, not provided by all the cards (see the Microsoft DDKs to see which functions are mandatory). If you use a facultative function, be careful to enclose it in an if statement to check the result.

Windows DDK provides a set of mandatory WLAN OIDs that should be supported by all Miniport drivers for IEEE 802.11; they are all defined in "ntddndis.h" file (from Windows XP SP1 DDK) and documented here.

The scan command is sent to the wireless card using the OID_802_11_BSSID_LIST_SCAN and the following function:

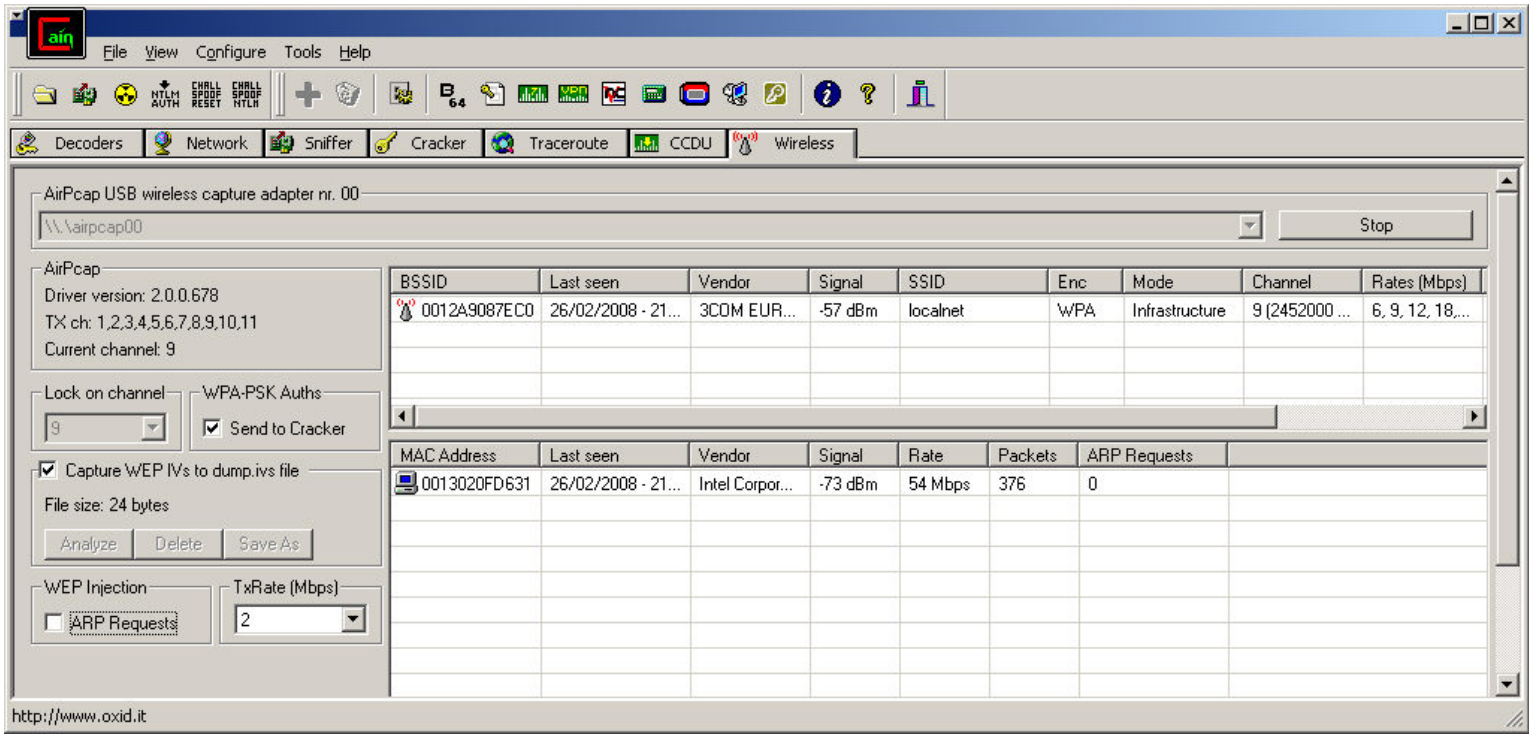
```
BOOL Packet_802_11_SetScanAP(LPADAPTER pAdapter)
{
	BOOL Status = FALSE;
	PPACKET_OID_DATA  pOidData;

	int len = sizeof(*pOidData);
	len += sizeof(UINT);

	pOidData = (PPACKET_OID_DATA) calloc(len, sizeof(char));
	pOidData->Oid = OID_802_11_BSSID_LIST_SCAN;
	pOidData->Length = sizeof(UINT);
```

```
        Status = PacketRequest(pAdapter,TRUE,pOidData); //Set
        free (pOidData);
        return Status;
}
```

Passive Scanner

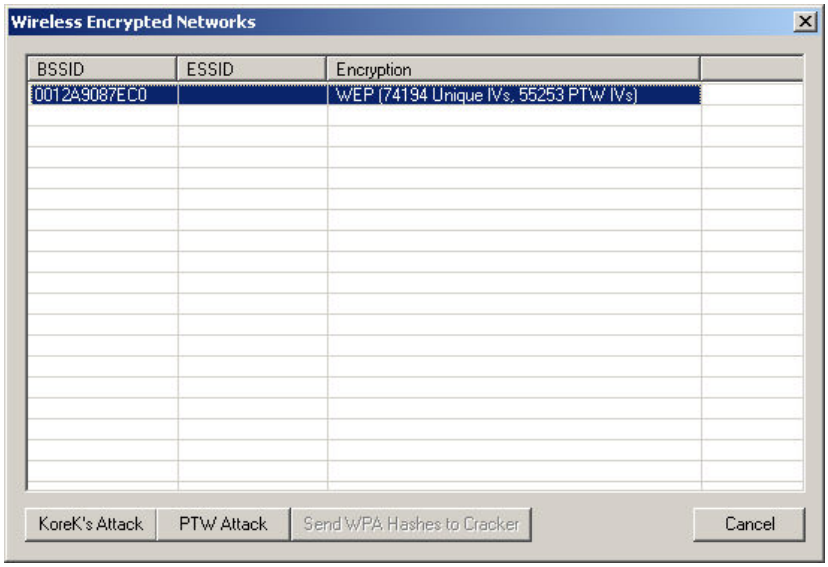
The passive scanner requires the AirPcap adapter from CACE Technologies which enables the raw capture of 802.11 frames by mean of its AirPcap drivers. The scanner recognize wireless Access Points (upper list) and clients (lower list) decoding 802.11b/g packets that travels on the air in a completely passive way. The "Channel Hopping" feature changes the frequency of the adapter every second and let you discover wireless networks on different channels.

When the "Dump WEP IVs" checkbox is checked, Cain collects unique WEP initialization vectors (IVs) in the "dump.ivs" file placed in the program's directory. WEP IVs are needed for cracking WEP encryption keys used in wireless protected networks.



Cain also support automatic ARP Requests injection (to speed up the collection of unique WEP IVs) and the capture of WPA-PSK authentication hashes. WEP injection is possible with specific Airpcap TX drivers only that can be obtained from CACE Technologies; you can check transmission enabled channels in the Airpcap driver information frame.

The WEP IVs dump file is compatible with those created by Aircrack-ng software. It can be opened immediately, using the "Analyze", button or saved for later analysis.



The above dialog lets you start Korek's Attack on WEP Keys; accordingly to Aircrack's documentation the minimum number of unique WEP IVs needed to successfully crack a WEP Key using the Korek's Attack is: 250.000 for 64-bit WEP keys and 1.000.000 or more for 128-bit keys.

You should be able to crack a 128-bit WEP key with 70.000 PTW IVs and the new PTW attack.

Requirements

This feature requires a Windows compatible wireless network interface and the Winpcap protocol driver. The Passive Scan feature requires the AirPcap adapter and drivers from CACE Technologies.

Usage

Choose the wireless adapter from the list and press the "Active Scan" or "Passive Scan" button.