The golden age of hacking

Covering tracks and hiding

Logs

Covert channels

Anatomy of an attack

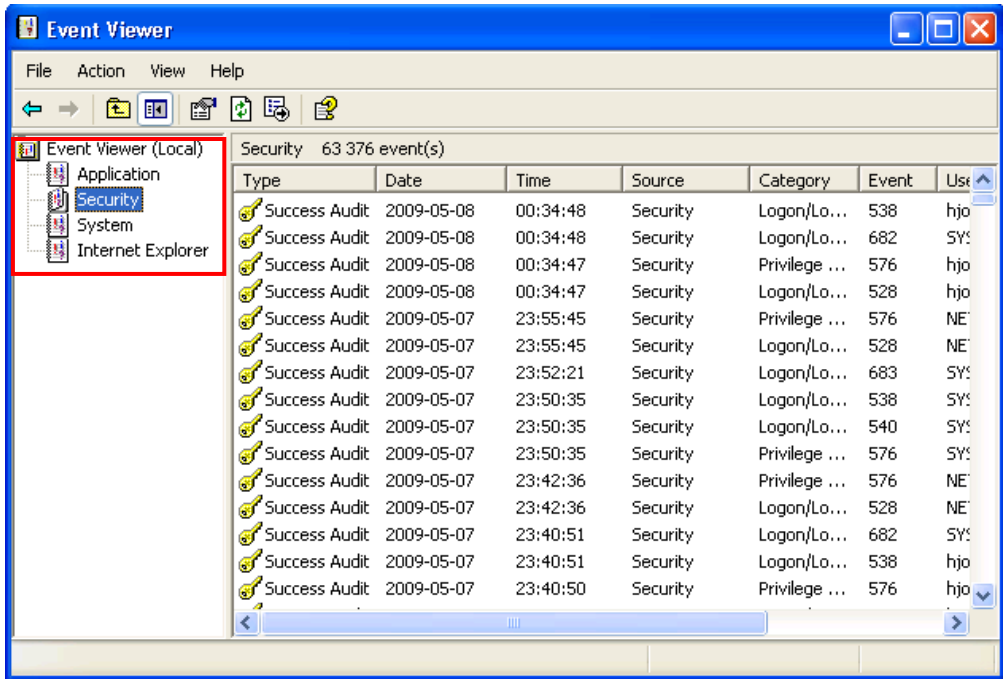
Covering tracks and hiding

- The majority of attacks are silent and stealthy
  - Elite attackers
- Sometimes attackers don't even need to hide
- Attractive home users
- Business networks is still the major target
- Hiding evidence by altering event logs
  - All attacks leaves traces!
  - What to delete?
- Logging to central server
- Things is a lot better with Vista and later Windows OS?
  - One improvement is that Windows 2003 security logs may now record the full IP address of machines attempting a login (previously only the NetBIOS name was recorded)

Read more about logging:
http://www.loganalysis.org/

Windows event logs (pre Vista)

- Logging is underused in most Windows networks
  - The most important - security log is turned off by default!
  - Mostly used when already been compromised
  - Usually a lot of data, hard to manage manually
  - http://www.windowsecurity.com/articles/Understanding_Windows_Logging.html
  - http://www.windowsecurity.com/articles/Understanding-Windows-Security-Templates.html
- Attacking event logs in Windows
  - The eventLog service produce a set of log files which ends with .LOG
  - They are periodically rewritten and event info are moved to the corresponding main log .EVT files
  - C:\WINDOWS\system32\config
  - Event viewer
  - Some MS apps have logs to



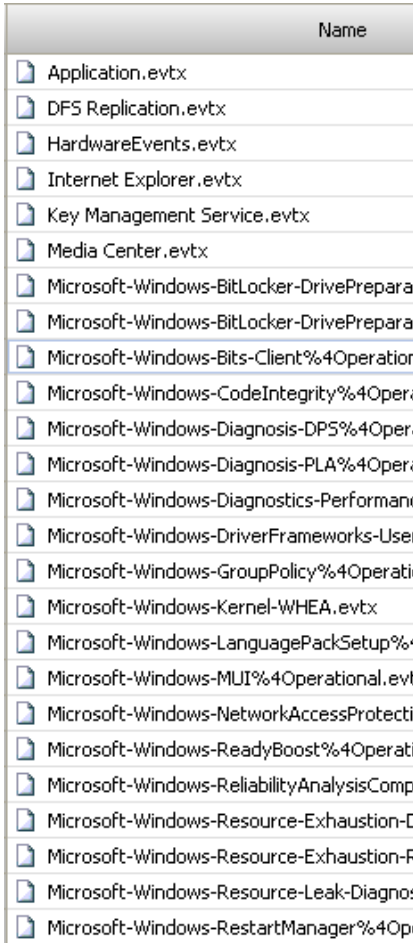
Vista/7 Event Logs

- The Windows event logs have changed dramatically in Windows Vista/7
  - A new **binary XML** file format is being used for the event logs with a new extension of .EVTX
- Log files are now located in
  - C:\Windows\System32\winevt\Logs\
- There are at least 30+ different event logs that Vista/7 report events to periodically
- Events can be forwarded and collected via subscriptions

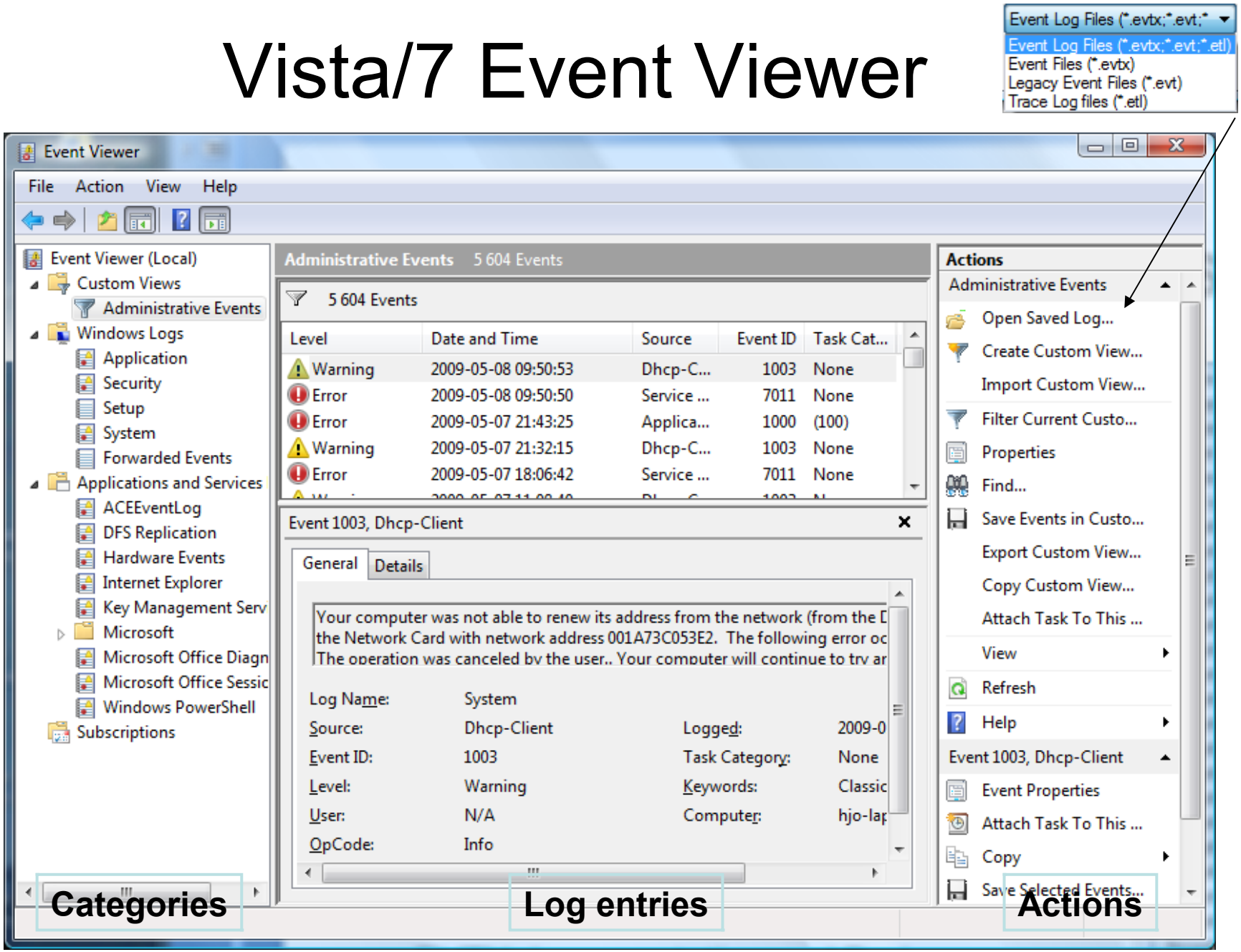
http://technet.microsoft.com/en-us/library/cc766042%28WS.10%29.aspx

- EVTX documentation and Perl parser

http://computer.forensikblog.de/en/topics/windows/vista_event_log/



Vista/7 Event Viewer



Altering event logs

- Altering Windows logs not easily possible on a live system
  - Binary format, owned and locked by Eventlog service
  - Stop Eventlog service, edit with proper access rights and special tools
  - Or boot from CD (physical access) – writing correct binary format
  - Winzapper (NT/W2K) or not public available tools?
    - http://www.securityfocus.com/tools/1726
- Altering Linux/Unix logs
  - /etc/syslog.conf (syslogd) tells where the logs are located (/var/log)
  - ASCII format – any text editor will do it
  - Need root or the same privilege as the daemon writing the log
  - Done by hand or by script
- utmp, wtmp, btmp and lastlog (w, who, last, lastb, lastlog, etc.)
  - Are binary files (utmp structure), lastlog may be distribution specific
  - http://www.packetstormsecurity.org/UNIX/penetration/log-wipers/

utmp.h structure (Ubuntu 9.04)

```
struct utmp {
        short   ut_type;              /* Type of record */
        pid_t   ut_pid;               /* PID of login process */
        char    ut_line[UT_LINESIZE]; /* Device name of tty - "/dev/" */
        char    ut_id[4];             /* Terminal name suffix, or inittab(5) ID */
        char    ut_user[UT_NAMESIZE]; /* Username */
        char    ut_host[UT_HOSTSIZE]; /* Hostname for remote login, or kernel version for run-level messages */
        struct  exit_status ut_exit;  /* Exit status of a process marked as DEAD_PROCESS;
                                         not used by Linux init(8) */
        /* The ut_session and ut_tv fields must be the same size when compiled 32- and 64-bit.
           This allows data files and shared memory to be shared between 32- and 64-bit applications. */
      #if __WORDSIZE == 64 && defined __WORDSIZE_COMPAT32
        int32_t ut_session;           /* Session ID (getsid(2)), used for windowing */
        struct {
            int32_t tv_sec;           /* Seconds */
            int32_t tv_usec;          /* Microseconds */
        } ut_tv;                      /* Time entry was made */
      #else
         long   ut_session;           /* Session ID */
         struct timeval ut_tv;        /* Time entry was made */
      #endif
        int32_t ut_addr_v6[4];        /* Internet address of remote host; IPv4 address uses just ut_addr_v6[0] */
        char __unused[20];            /* Reserved for future use */
      };
```

Altering history files and defense

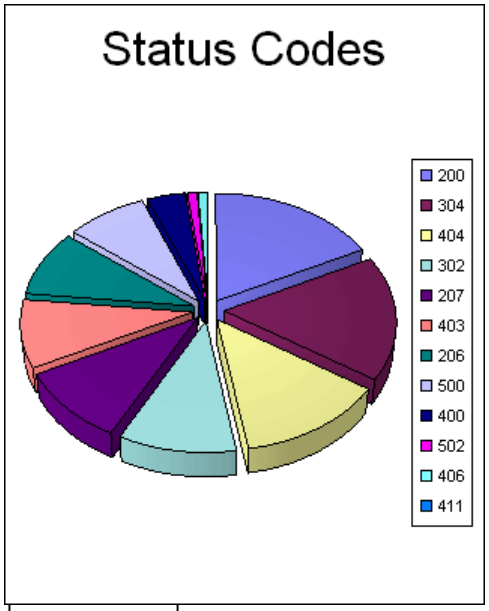
- /home/user/.shellName_history
  - Stores the 500 last entered commands
  - However the last commands are not written until shell is exited
    - Kill the current shell instead of logout
  - /home/user/ may have some other history dirs/files as .mc/ .lesshst etc.
- Activate logging
  - Not a problem in Unix/Linux – logrotate
    - Rotate, compress (and mail logs), run as a daily cron job
  - Windows XP – most is off
    - Windows default is 512kB - change in Event Viewer properties
    - Needs thorough auditing review in security policy settings
- Additional log file protection
  - Proper permissions, append only, encrypted…
    - http://www.coresecurity.com/ Corelabs > Open Source Projects > MSyslog
  - Separate secured logging server with local logging still on
  - Attached log box (no network) or write once media as DVD-R

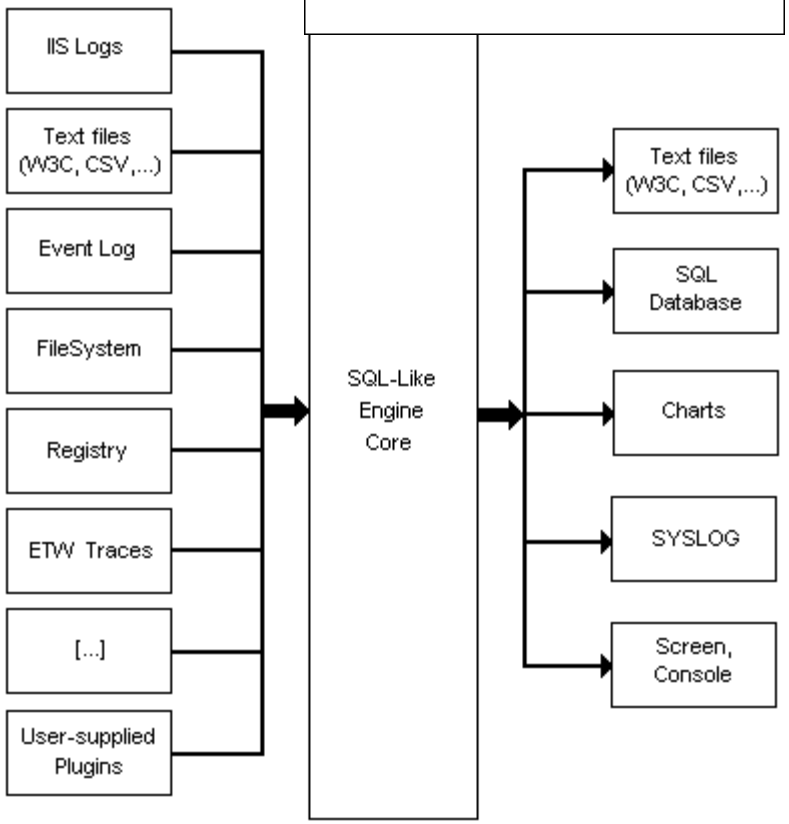
MicroSoft Log Parser (free)

- As an application developer you often need to write some logs for your application
  - There is many logging framework to choose among: Log4net, Log4j, Microsoft Logging Application Block, etc.
  - But when it come to read those logs, search for data, create reports, extract statistics or perform some alert/action on them, things become harder
- Log Parser performs SQL queries against a variety of log files and other system data sources
  - You can query any log and data sources (database, event log, IIS logs, file system, registry, etc.) with a complex SQL query!
  - On the down side, using it from the command line become quickly unpractical as you need to type your SQL query in a DOS prompt
    - logparser -i:EVT "SELECT TOP 20 * FROM Security WHERE EventID=5032 ORDER BY TimeGenerated DESC" -o DATAGRID
    - logparser -i:W3C -o:DATAGRID ”SELECT RowNumber, date, time, action, protocol, src-ip, dst-ip, src-port, dst-port FROM C:\cases\pfirewall.log WHERE dst-port IN (80; 443) ORDER BY RowNumber”

Log Parser Architecture

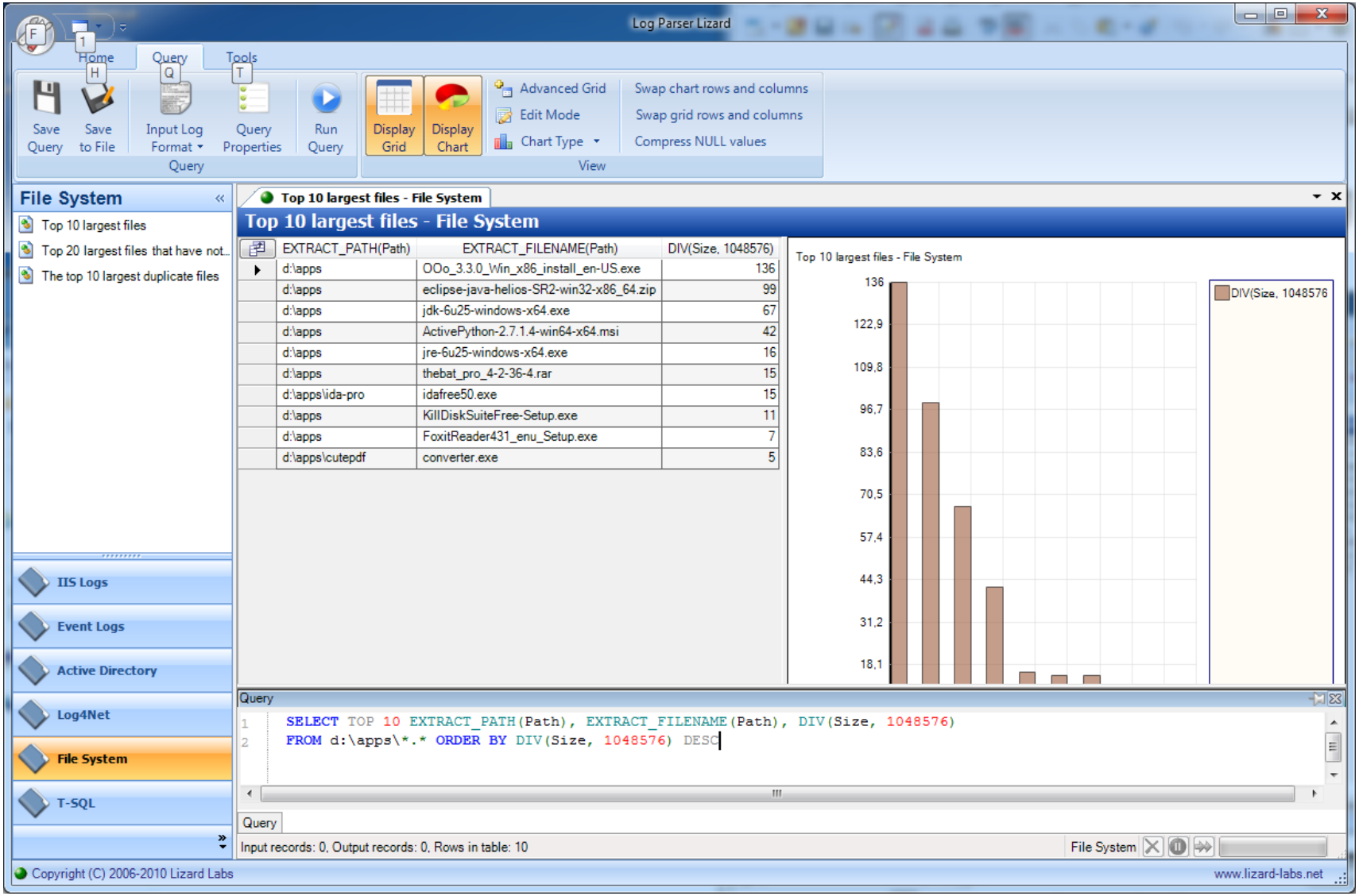
- Swiss Army knife for processing Windows logs of all types (and others). The world is your database with Log Parser!
- **Input Formats** are generic *record providers*
  - Input Formats can be thought of as SQL tables containing the data you want to process
  - Manage .evtx files as well
- A **SQL-Like Engine Core** processes the records generated by an Input Format
  - SQL language (SELECT, WHERE, GROUP BY, HAVING, ORDER BY etc.)
  - Aggregate functions (SUM, COUNT, AVG, MAX, MIN etc.)
  - A rich set of functions (e.g. SUBSTR, CASE, COALESCE, REVERSEDNS, etc.)
- **Output Formats** are generic *consumers of records*
  - They can be thought of as SQL tables that receive the results of the data processing
  - BSD syslog protocol, RFC 3164



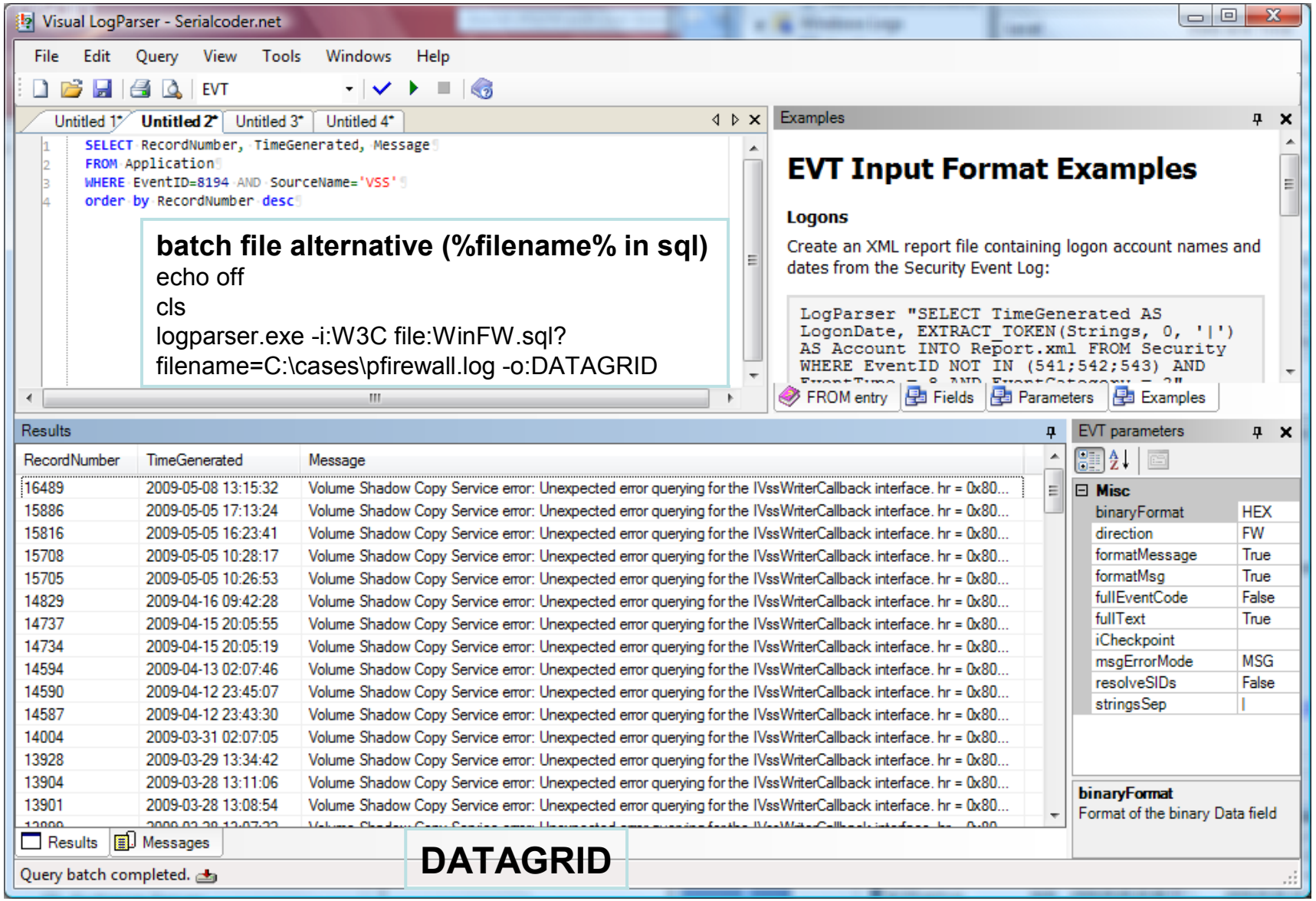


Log Parser Lizard

http://www.lizard-labs.net/log_parser_lizard.aspx



SQALP (Simple Query Analyzer for Log Parser)



MicroSoft Log Parser, events etc.

- Log Parser download
  - http://www.microsoft.com/technet/scriptcenter/tools/logparser/default.mspx
- Visual Log Parser GUI (SQALP)

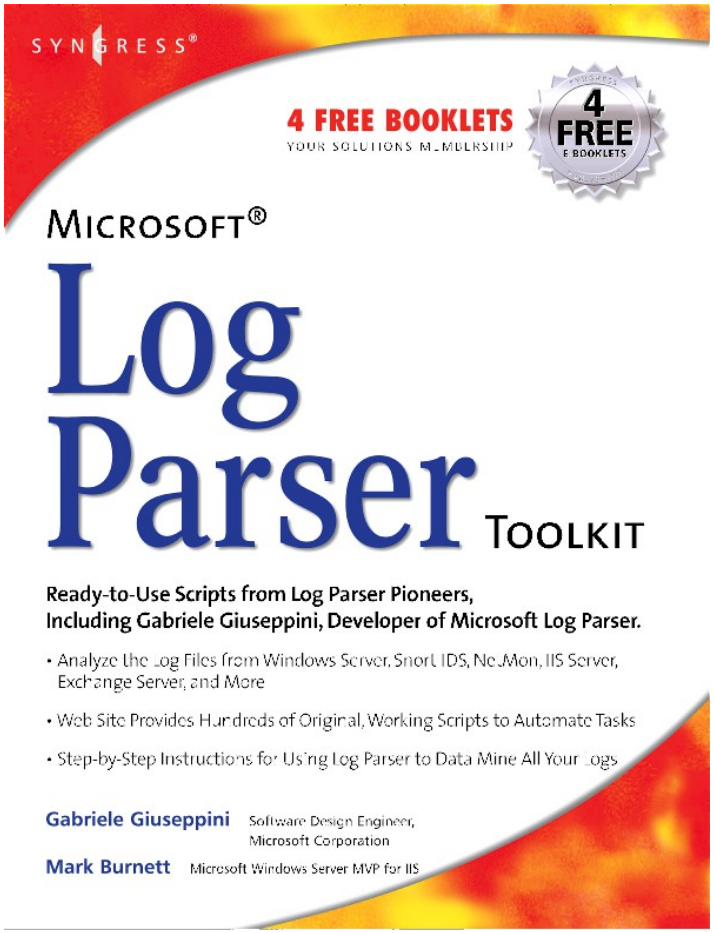
http://en.serialcoder.net/logiciels/visual-logparser.aspx

- Log Parser user forum
  - www.logparser.com
- Book with loads of scripts and queries

http://www.elsevierdirect.com/companion.jsp?ISBN=9781932266528

- Microsoft log events
  - http://eventlogs.blogspot.com
  - http://eventid.net (what does it mean?)
- Forensic Log Parsing with Microsoft's Log Parser
  - http://www.securityfocus.com/infocus/1712

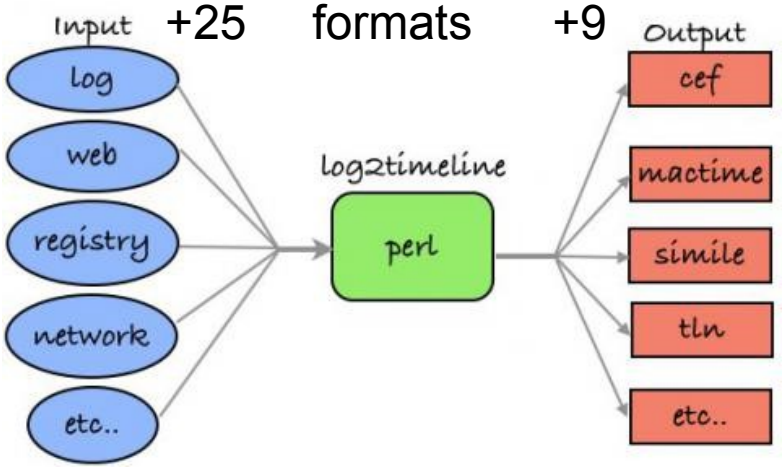
"Mastering Windows Network Forensics and Investigation" have a good tutorial as well!

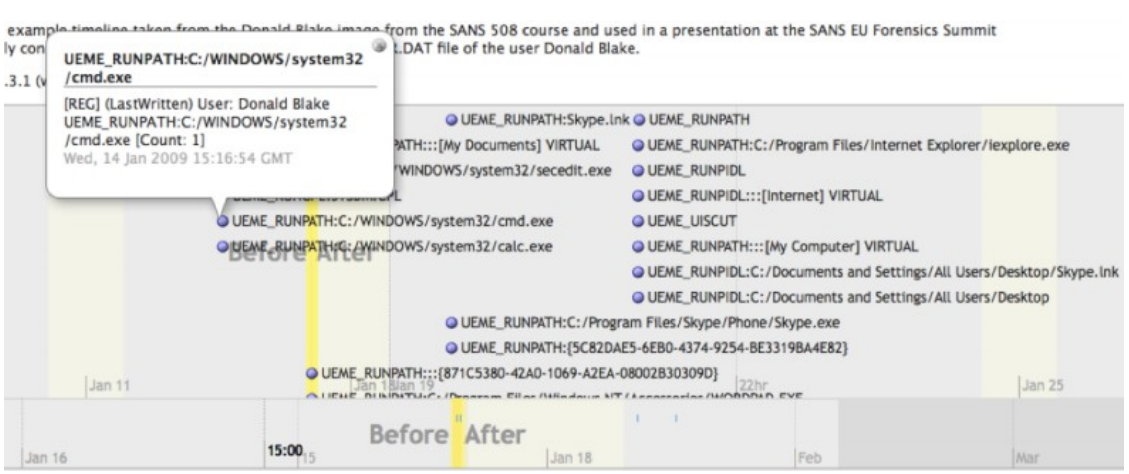


Log2timeline - http://log2timeline.net/

- A framework for automatic creation of a super timeline. The main purpose is to provide a single tool to parse various log files and artifacts found on suspect systems (and supporting systems, such as network equipment) and produce a timeline that can be analysed by forensic investigators/analysts
- The tool is written in Perl for Linux but has been tested using Mac OS X (10.5.7+ and 10.6.+). Parts of it should work natively in Windows as well (with ActiveState Perl installed)
- "Mastering the Super Timeline With log2timeline" can be downloaded here
  - http://www.sans.org/reading_room/whitepapers/logging/mastering-super-timeline-log2timeline_33438

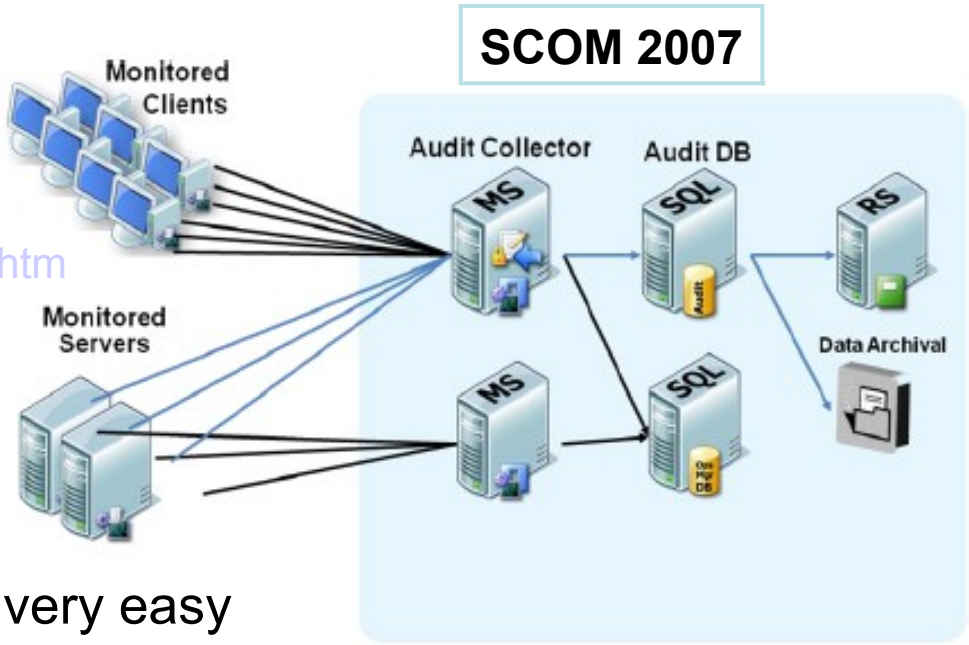
SIMILE: http://www.simile-widgets.org/timeline/





Microsoft System Center Operations Manager 2007 R2 and Syslog (RFC 3164) alternatives

- Microsoft System Center Operations Manager är ett händelse- och prestandaövervakningsverktyg som innehåller en mängd funktioner för att reducera den tid det tar att konfigurera ett system eller en tillämpning
- Course and other white papers
  - http://www.microsoft.com/systemcenter/operationsmanager/en/us/default.aspx
- End-to-End Service Monitoring
- Client Monitoring
- Audit Collection



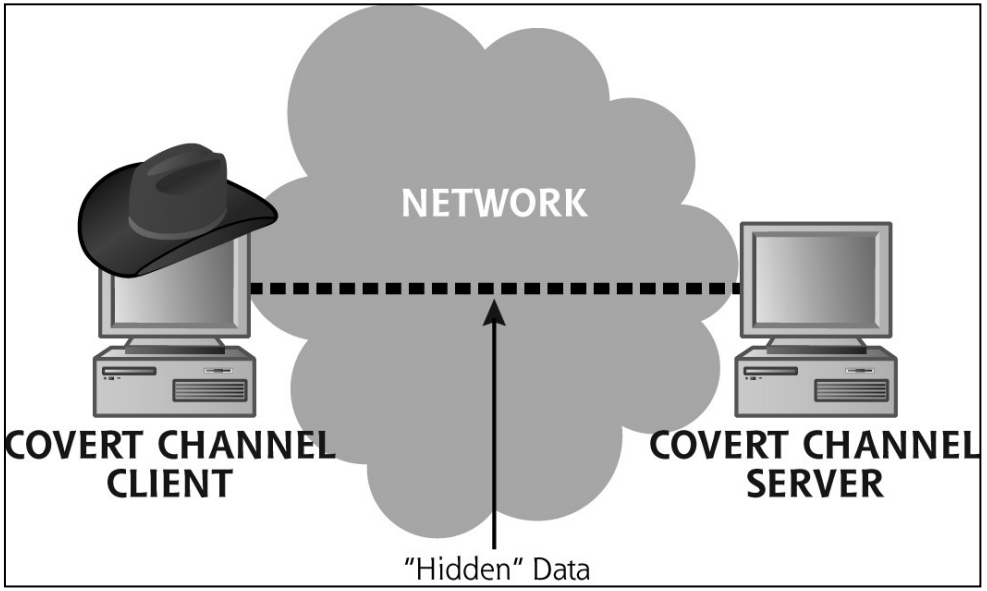
- GNU/Linux setup
  - http://www.aboutdebian.com/syslog.htm
- Other (Windows) Syslog servers
  - http://en.wikipedia.org/wiki/Syslog
- Convert Windows log to Syslog
  - http://www.syslogserver.com
- Setting up Syslog to redirect logging to separate log server is very easy

Hiding files and directories

- Unix/Linux
  - '.' = current directory, '..' = parent directory (already present)
  - '.' and different combinations hides files and folders
    - Ex. [dot] [dot] [space] or [dot] [dot] [dot] etc.
  - ls -al check "man ls" for more options
- Windows
  - Hidden attribute
  - ADS stream (only NTFS)
    - Create: C:\tmp>type malware.exe > my.txt:hidden-malware.exe omitting "my.txt" attach ADS stream to the tmp folder instead
    - Restore: C:\tmp>more < my.txt:hidden-malware.exe > malware.exe
    - Run: C:\tmp>start .\my.txt:hidden-malware.exe
  - Other places
    - C:\System Volume Information, System.sav and Recycler folders
    - MSOCache and other obscure hidden places
    - Hide protected operating system files
  - Metasploit MAFIA
    - Slacker tool
- Make sure AV and other scan tools are ADS aware
  - Otherwise same defense as with rootkits etc.

Covert channels I

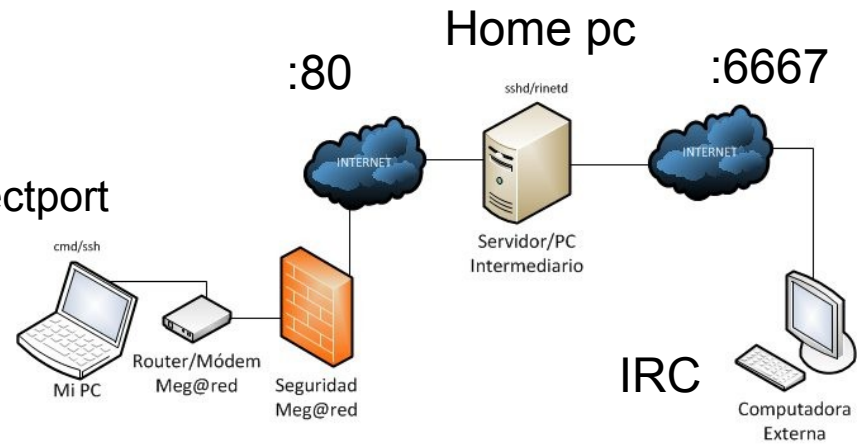
- Hide data on the network
- Installed via
  - Some vulnerability
  - E-mail trojan
  - Ex-employee
  - Contractor or temp
  - Physical break-in
- Port redirection
- Tunneling
  - Allows encapsulation of any protocol within another enabling authorized data streams to carry arbitrary data
  - SSH forward and reverse tunnels, SSL wrapping via Stunnel
  - HTTP tunneling and third party VPN services
  - http://www.chrisbrenton.org/2009/08/top-5-firewall-threats/



Port redirection with rinetd etc.

- Port redirection involves accepting traffic on a network interface, on a specific port, and redirecting it to a different IP address/port. This ability can be useful in several situations
- Imagine you are at a office which is protected by a firewall with strict outbound rules, allowing only outbound traffic on port 80 (no content inspection)
- You are an IRC addict and must constantly be connected to your favourite IRC server in order maintain your mental health :)
- On your home computer you can listen on port 80 and redirect any incoming traffic to to the IRC server at port 6667
- There are several port redirectors for Windows as FPipe and WinRelay. Unix got the internet redirection server Rinetd, which is present in BackTrack
- We can configure Rinetd using /etc/rinetd.conf

```
# allow 192.168.2.*
# deny 192.168.2.1
# forwarding rules come here
# bindaddress    bindport  connectaddress  connectport
130.64.228.230  80  irc.freenode.net  6667
# logging information
logfile /var/log/rinetd.log
```



Port forward SSH tunnel

Dynamic and Local

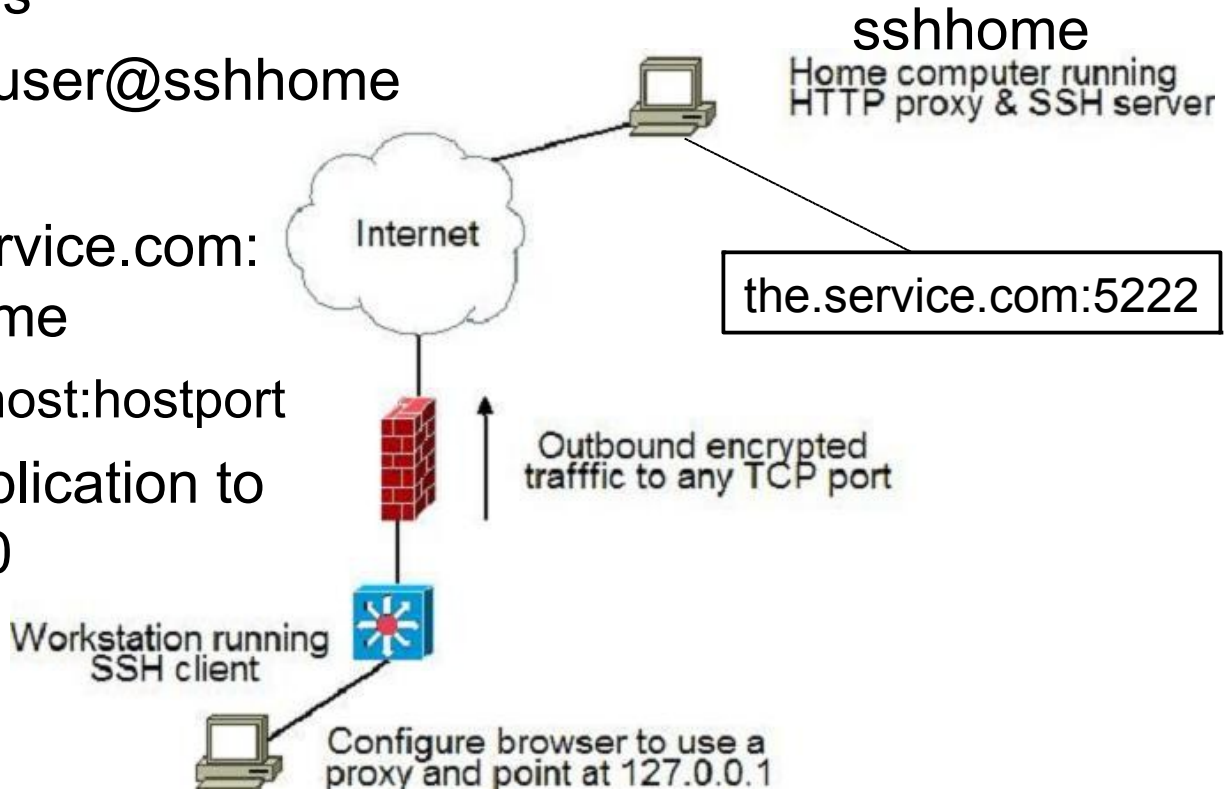
- SSH tunnel sessions manage to encrypt traffic and create bi-directional channels which can be used to forward local and remote connections
- This feature allows one to do seemingly impossible TCP/UDP traffic manipulations
- putty.exe -D 1080 user@sshhome

- D [bind_address:]port

- ssh -L 3000:the.service.com: 5222 user@sshhome

-L [bind_address:]port:host:hostport

Config local client application to use 127.0.0.1:3000



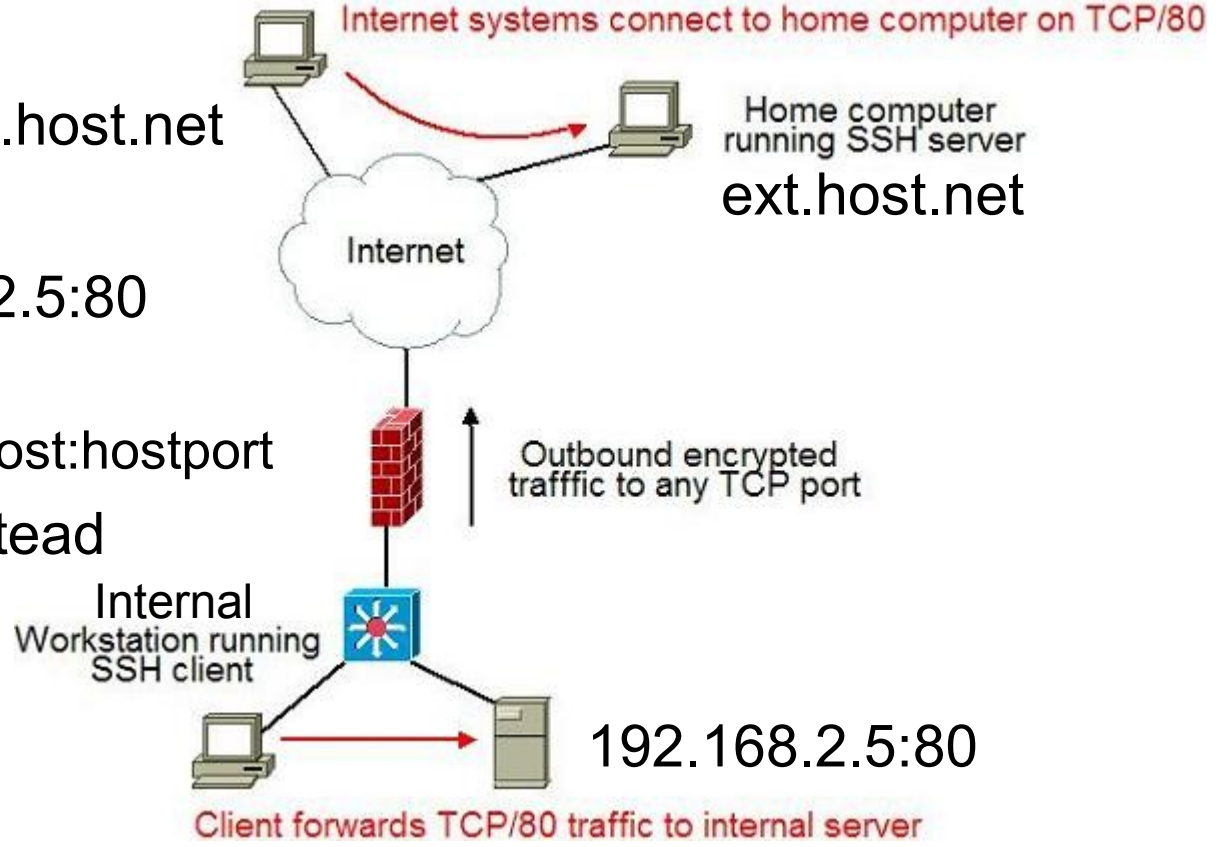
Reverse port forward SSH tunnel

Remote server is forwarded to local client

- Let's say we have an internal server which is only accessible to internal employees
- And we have an end user who wishes to expose this server to the Internet
- **External**
- ssh -p 80 user@ext.host.net
- **Internal**
- ssh -R 80:192.168.2.5:80 user@ext.host.net

-R [bind_address:]port:host:hostport

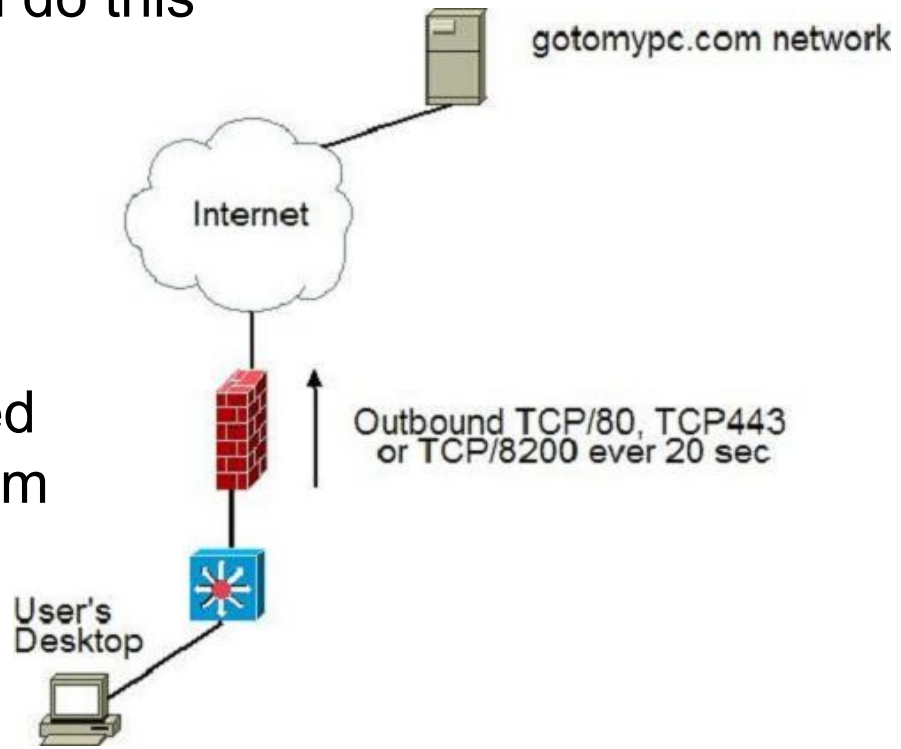
- Using 127.0.0.1 instead we could redirect traffic to a port on the internal client



Tunneling SSH over HTTP(S) and third party VPN

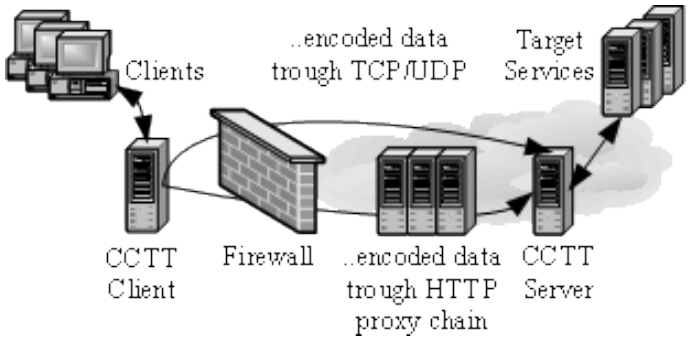
- Enables the user to run SSH connections over most HTTP and HTTPS proxy servers. Due to SSH features such as port forwarding, this can allow many types of services to be run safely over the SSH via HTTP connections
- Several proxy servers are supported & Apache via mod proxy
- Corkscrew and Proxytunnel can do this

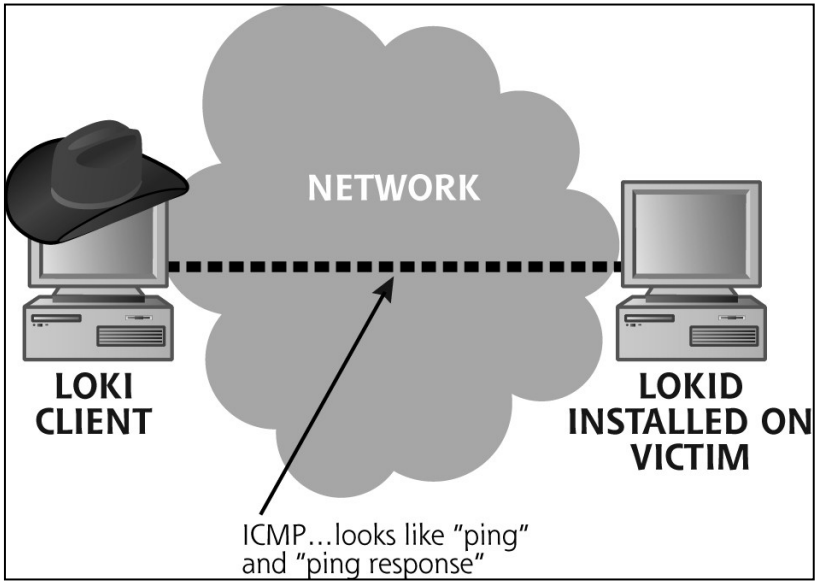
- Third party VPN services allow end users to create an encrypted tunnel between their work system and systems on the Internet
- OpenVPN – configured virtual servers are downloadable

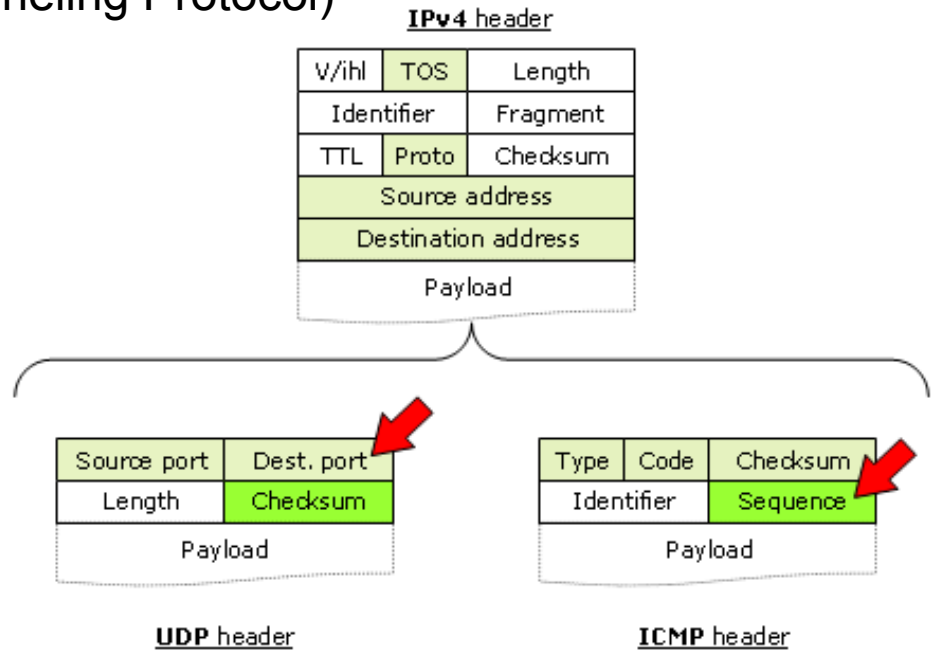


Covert channels II

- Loki use ICMP as tunnel
  - All traffic is wrapped in ICMP payload field
  - Extracts the packets from the kernel
  - Port scans are futile
  - Also supports encryption and UDP port 53
- Others examples from: http://www.gray-world.net/
  - CCTT (The Covert Channel Tunneling Protocol)
  - MSNShell, Firepass (HTTP), …

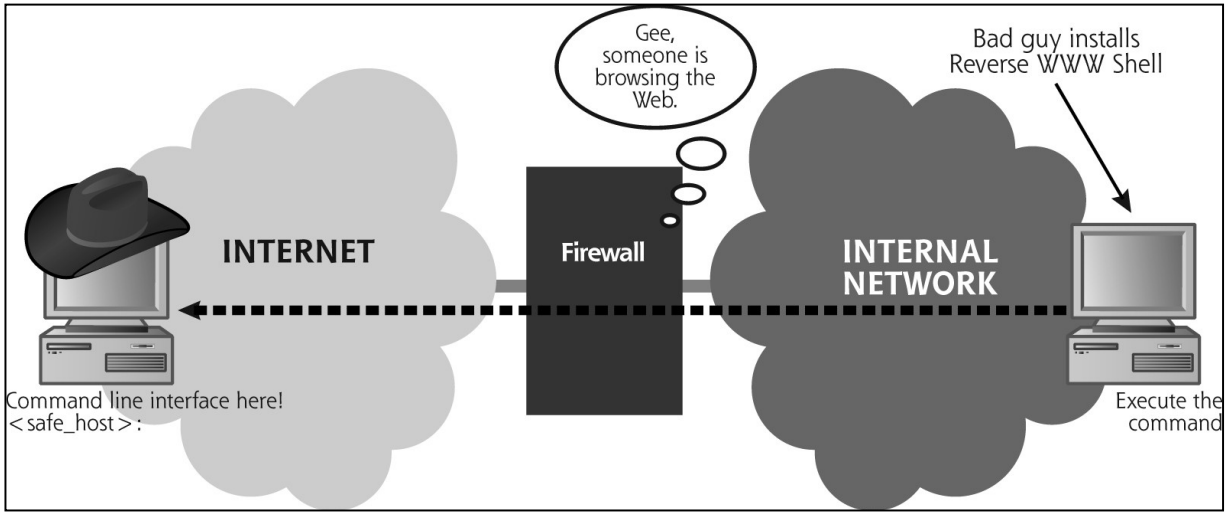




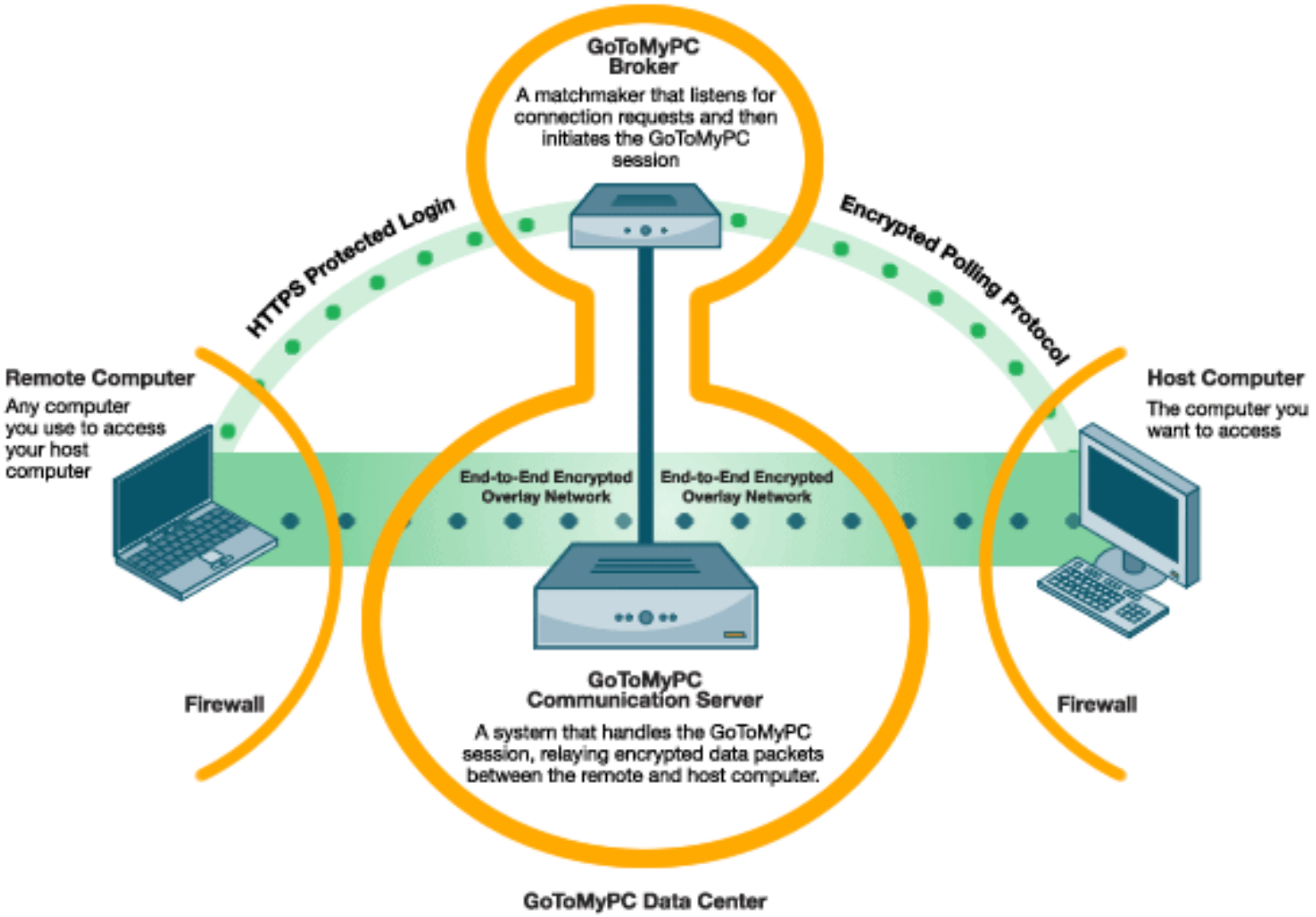


Covert channels III

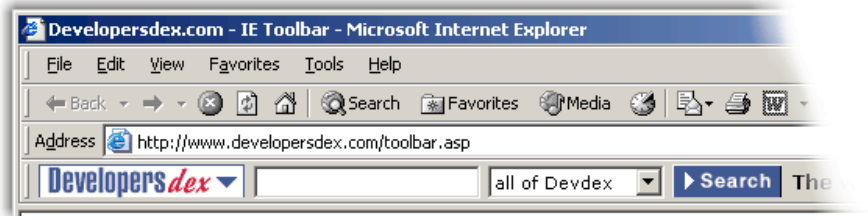
- Reverse WWW shell (as PassiveX)
  - Carry shell commands in standard HTTP GET messages
  - Victim appears to surf the web
    - Randomly polls the attacker for new commands to execute
  - Attacker appears to be a WWW server
  - Support even user/pass for the outgoing web proxy firewall
  - Similar to remote services as: https://www.gotomypc.com
- Other implementations
  - SMTP (slow)
  - FTP
  - Streaming Audio
  - TCP/CP (Carrier Pigeon) ☺



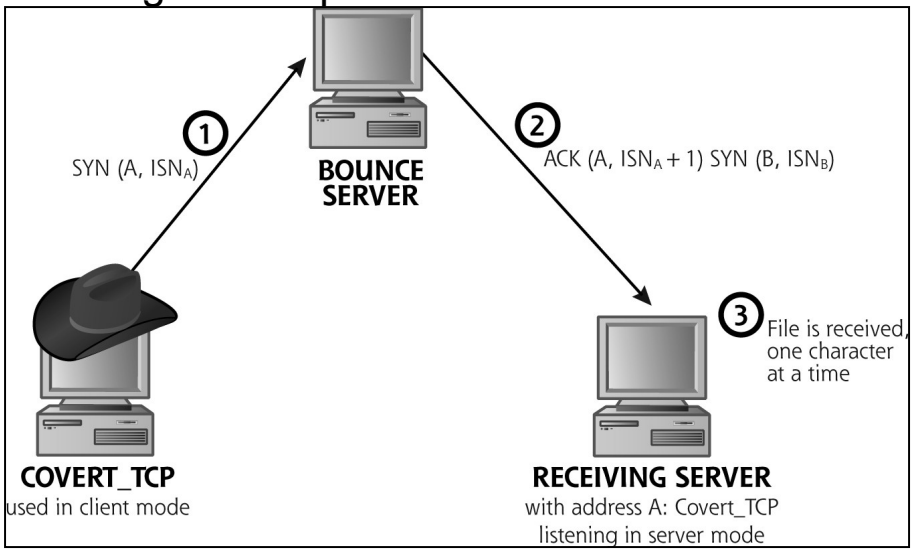
http://www.gotomypc.com/remote_access/remote_access_technology



Even more covert channels I

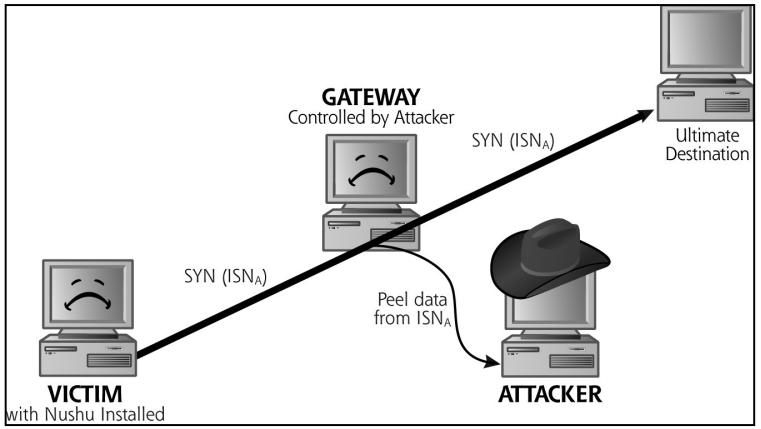


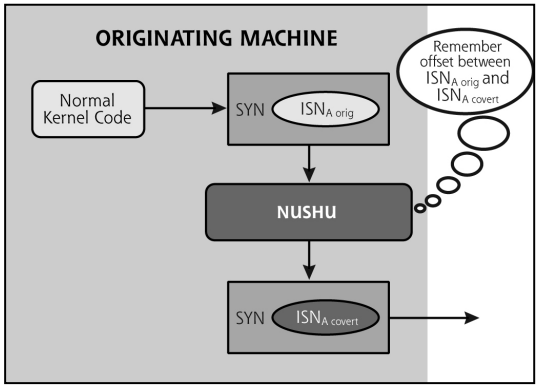
- Mal/spy -ware using web browser
  - Piggyback IE using HTTP/HTTPS
  - BHO (Browser Helper Object)
    - http://en.wikipedia.org/wiki/Browser_Helper_Object
    - List BHOs: BHODemon and bho.pl (Windows Forensic Analysis)
- Using TCP and IP headers to carry data
  - Covert_TCP
  - Simple transfer of one char at a time using either one of the fields
    - IP ID (identification), usually used with fragmented packets…
    - TCP sequence number
      - Then RESET
    - TCP acknowledgment number
      - Used with bounce operation
  - Bounce operation
    - Spoofed source IP address
    - Send value of char-1 as ISNa
    - Works even if dest. port is closed on BNC (RESET have ISNa)

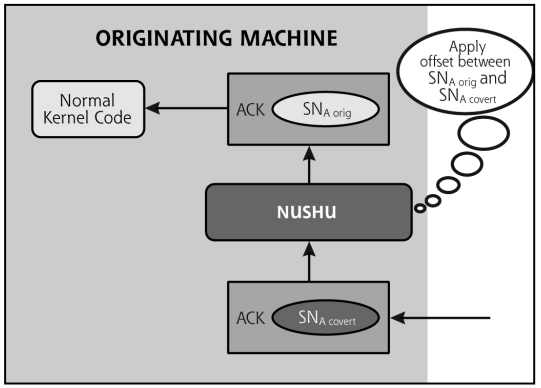


Even more covert channels II

- Noshu
  - Passive covert channel
  - Insert and peel data from ISNa during three-way handshake by other applications
  - Both victim and gateway must be under control!
  - Keep track of sequence numbers
- Defense
  - As usual no root/admin access
  - Hardened OS
  - Patched
  - Anti-virus and anti-spyware software
  - Unusual processes?
  - IDS systems as Snort

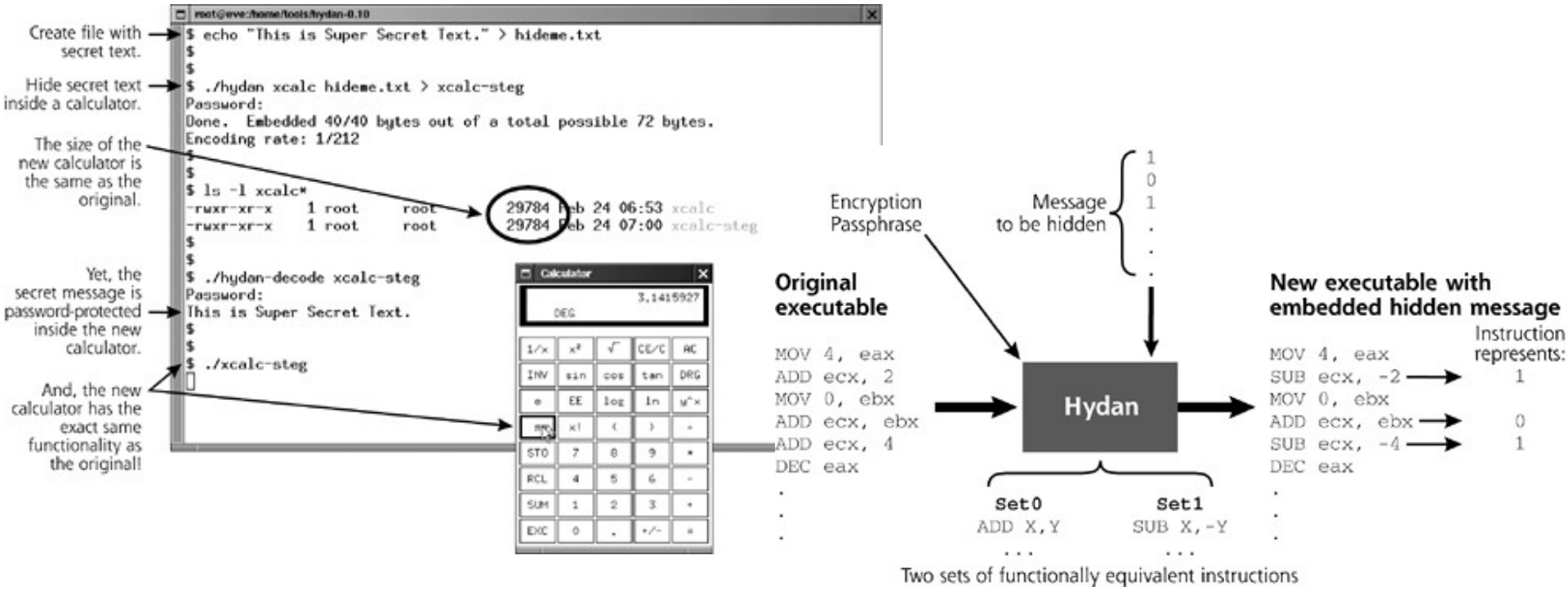






Steganography Techniques: Embedding in binary files

- Hiding info in images etc. in local files, web servers etc.
- Hydan
  - Exactly the same size and function but new hash sum
  - http://www.crazyboy.com/hydan/



Anatomy of an attack

- Most of the attacks is done with steps as in the CHR book according to author
  - But there is a lot of difference depending on
    - Circumstances
    - Skill
    - Tools
    - Time etc.
- CHR book have three scenarios
- PDF from Handbok i IT-säkerhet
  - lab2\netbios\bilaga_om_hackning.pdf

IP and TCP headers