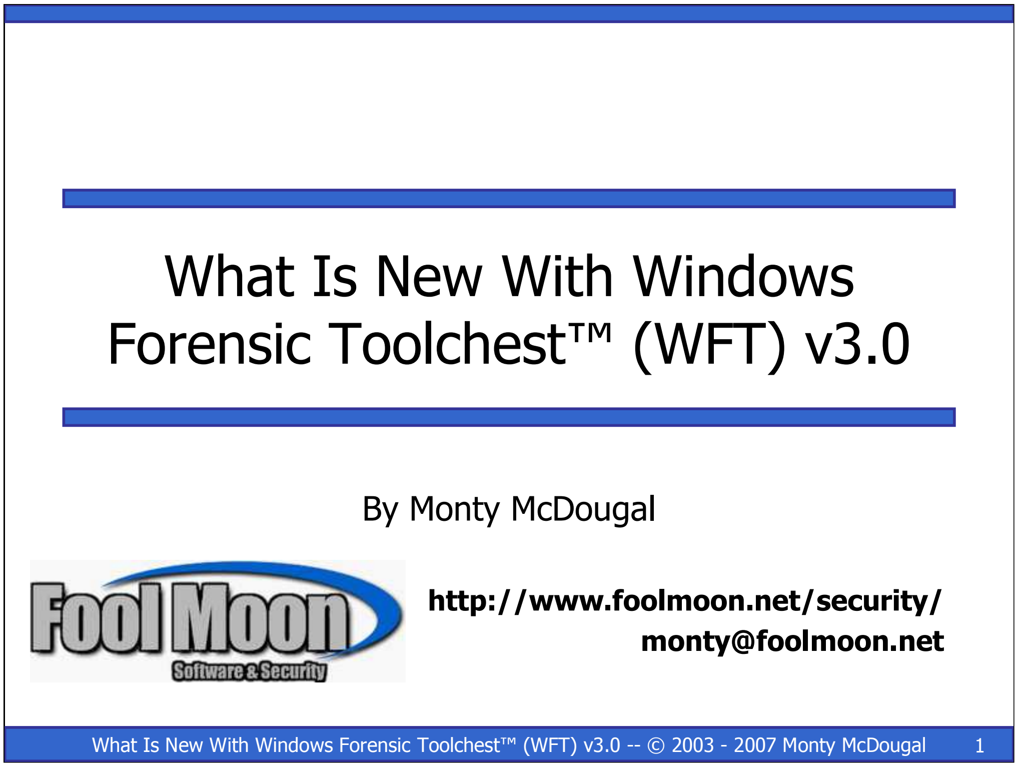


# What Is New With Windows Forensic Toolchest™ (WFT) v3.0

By Monty McDougal

**http://www.foolmoon.net/security/**
**monty@foolmoon.net**

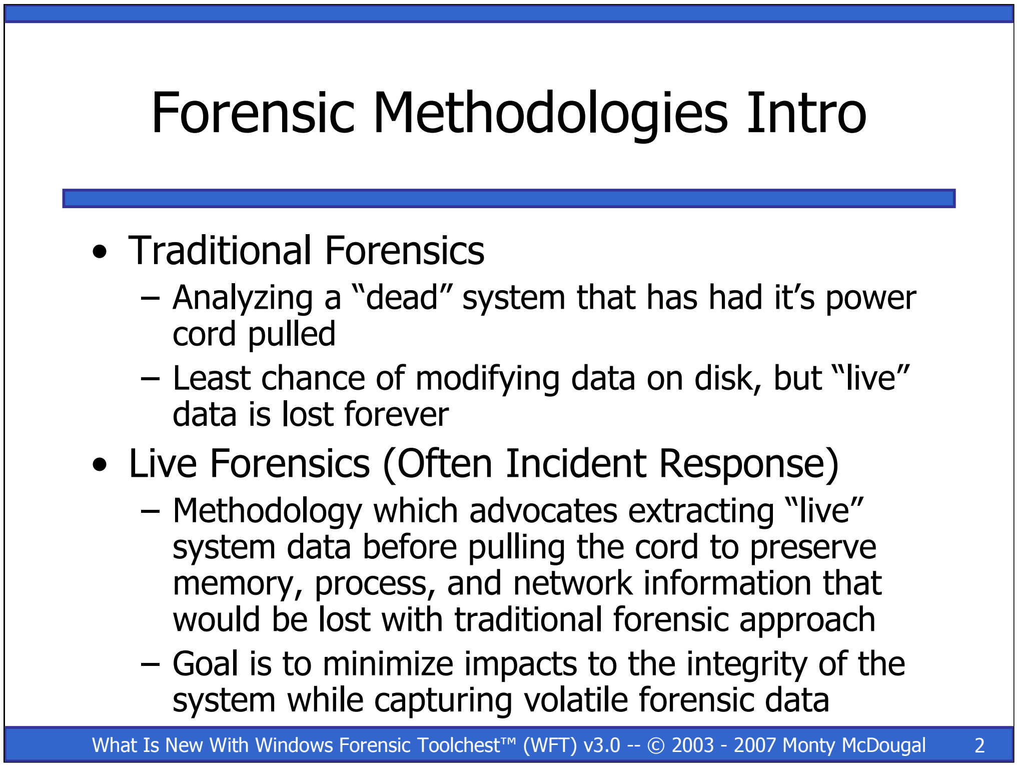
What Is New With Windows Forensic Toolchest™ (WFT) v3.0 -- © 2003 - 2007 Monty McDougal 1

**What Is New With Windows Forensic Toolchest™ (WFT) v3.0**

**By Monty McDougal**

**http://www.foolmoon.net/security/**

**monty@foolmoon.net**



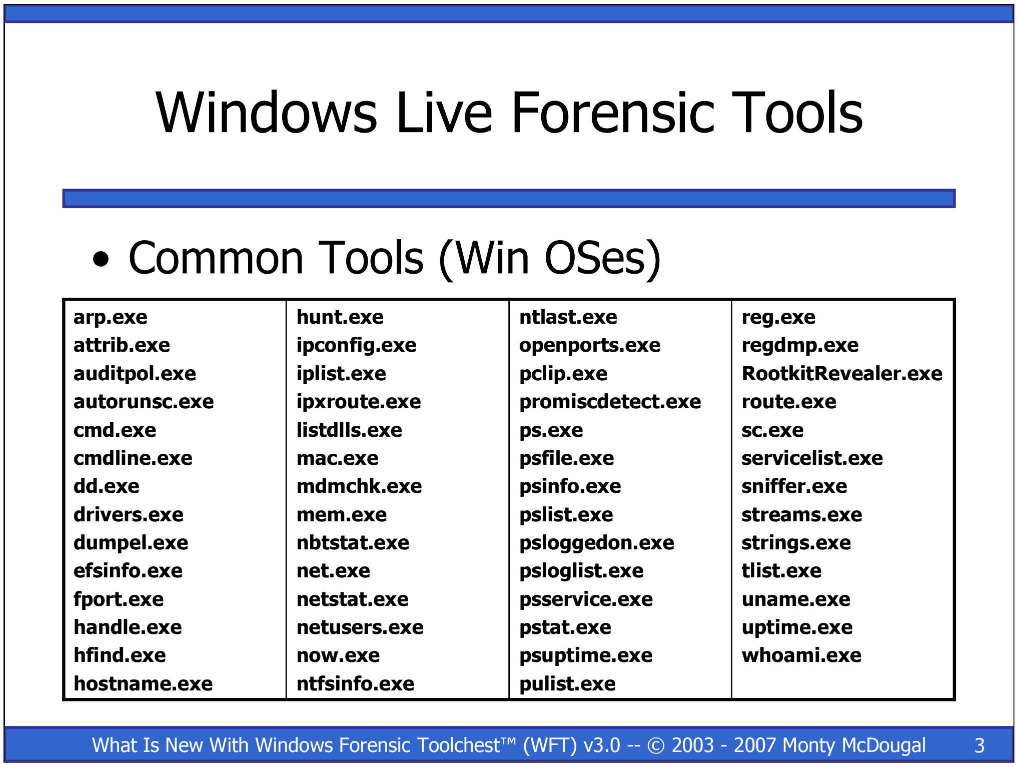
## Forensic Methodologies Intro

- Traditional Forensics
  - Analyzing a “dead” system that has had it’s power cord pulled
  - Least chance of modifying data on disk, but “live” data is lost forever
- Live Forensics (Often Incident Response)
  - Methodology which advocates extracting “live” system data before pulling the cord to preserve memory, process, and network information that would be lost with traditional forensic approach
  - Goal is to minimize impacts to the integrity of the system while capturing volatile forensic data

Forensic methodologies, generally fall into two broad camps.

The first is the “pure” pull-the-plug traditional forensic methodology advocated for many years by most of the law enforcement community. This method is great for preserving data on disk, but you lose allot of volatile data which may be useful. A skillful attacker may never even write their files to disk. A real world example of this is the code red worm.

The second methodology, live forensics, recognizes the value of the volatile data that may be lost by a power down and seeks to collect it from a running system. As any such action will in some minor ways later the system, it is not pure in forensic terms. Many people, including the author of this presentation, feel this is an acceptable tradeoff given the value of the data that can be collected from a running system (with minimal impacts).



## Windows Live Forensic Tools

- Common Tools (Win OSes)

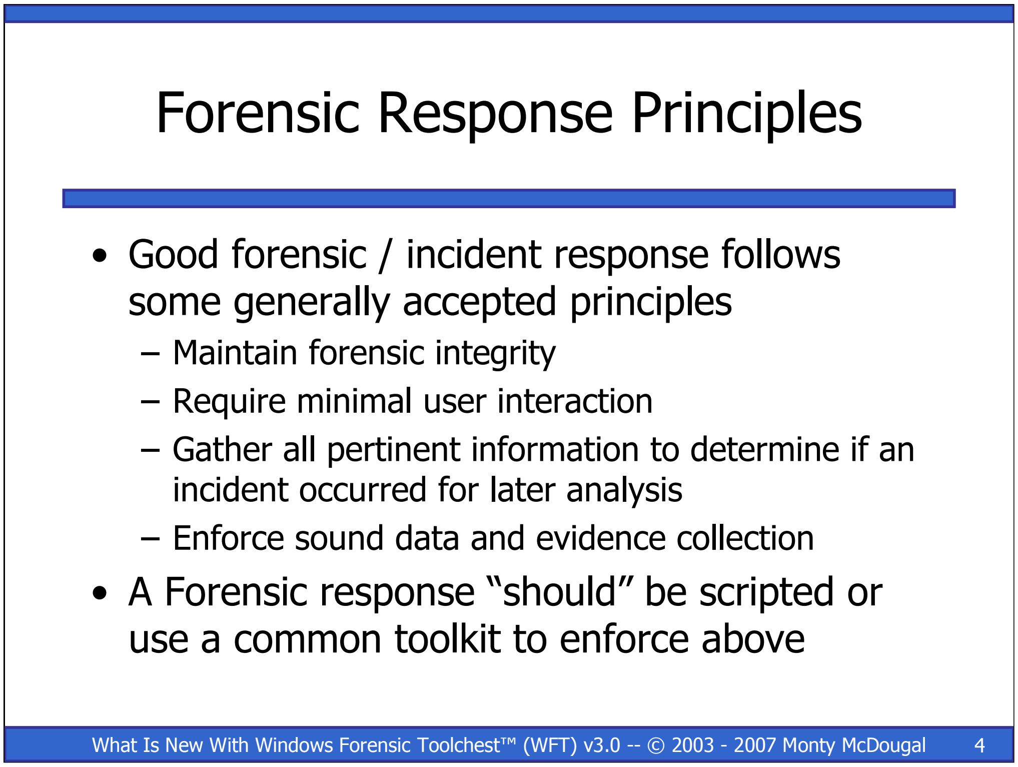
**arp.exe**<br>
**attrib.exe**<br>
**auditpol.exe**<br>
**autorunsc.exe**<br>
**cmd.exe**<br>
**cmdline.exe**<br>
**dd.exe**<br>
**drivers.exe**<br>
**dumpel.exe**<br>
**efsinfo.exe**<br>
**fport.exe**<br>
**handle.exe**<br>
**hfind.exe**<br>
**hostname.exe**

**hunt.exe**<br>
**ipconfig.exe**<br>
**iplist.exe**<br>
**ipxroute.exe**<br>
**listdlls.exe**<br>
**mac.exe**<br>
**mdmchk.exe**<br>
**mem.exe**<br>
**nbtstat.exe**<br>
**net.exe**<br>
**netstat.exe**<br>
**netusers.exe**<br>
**now.exe**<br>
**ntfsinfo.exe**

**ntlast.exe**<br>
**openports.exe**<br>
**pclip.exe**<br>
**promiscdetect.exe**<br>
**ps.exe**<br>
**psfile.exe**<br>
**psinfo.exe**<br>
**pslist.exe**<br>
**psloggedon.exe**<br>
**psloglist.exe**<br>
**psservice.exe**<br>
**pstat.exe**<br>
**psuptime.exe**<br>
**pulist.exe**

**reg.exe**<br>
**regdmp.exe**<br>
**RootkitRevealer.exe**<br>
**route.exe**<br>
**sc.exe**<br>
**servicelist.exe**<br>
**sniffer.exe**<br>
**streams.exe**<br>
**strings.exe**<br>
**tlist.exe**<br>
**uname.exe**<br>
**uptime.exe**<br>
**whoami.exe**

Above is a list of the most common tools used in a forensic response on a windows system. Most of these are either free downloads or come from Microsoft as part of their OS / Resource kits. The location to acquire these is all documented within the WFT config file. A few have been deprecated and may be difficult to locate, but that is also noted in the config file.



## Forensic Response Principles

- Good forensic / incident response follows some generally accepted principles
  - Maintain forensic integrity
  - Require minimal user interaction
  - Gather all pertinent information to determine if an incident occurred for later analysis
  - Enforce sound data and evidence collection
- A Forensic response “should” be scripted or use a common toolkit to enforce above

Like most things, there is generally a “right way” and a “wrong way” to do live forensics. The “right way” will generally exhibit four traits:

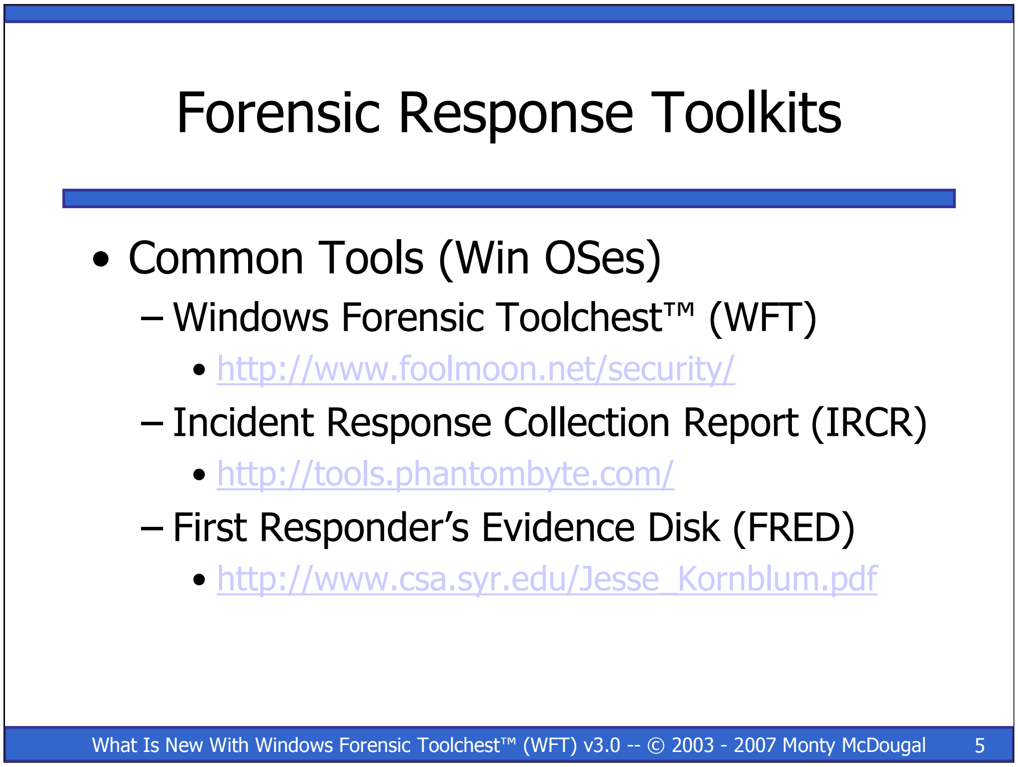
•Maintain forensic integrity

•Require minimal user interaction

•Gather all pertinent information to determine if an incident occurred for later analysis

•Enforce sound data and evidence collection

One of the keys to any such response, is that it be consistent and verifiable. Therefore, it is highly recommended that the response be automated. There are a number of common toolkits which will assist with this on a windows system. These are covered on the next slide.

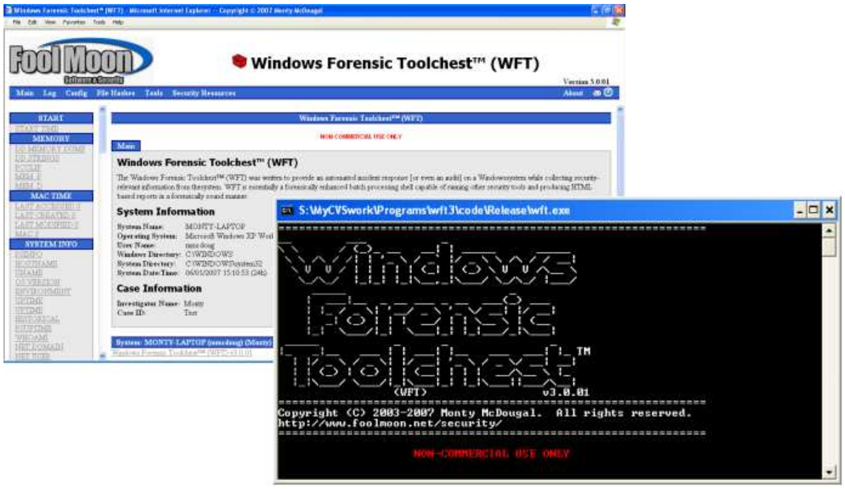


## Forensic Response Toolkits

- Common Tools (Win OSes)
  - Windows Forensic Toolchest™ (WFT)
    - http://www.foolmoon.net/security/
  - Incident Response Collection Report (IRCR)
    - http://tools.phantombyte.com/
  - First Responder's Evidence Disk (FRED)
    - http://www.csa.syr.edu/Jesse_Kornblum.pdf

These are probably three most common forensic response toolkits for Windows. I am of course biased towards Windows Forensic Toolchest™ (WFT). IRCR has been completely rewritten (borrowing from WFT) and is considerably more powerful than it was when I wrote my original paper and WFT v1.0. If you are looking for alternative to WFT, this is the one I would recommend. FRED, has a slightly different goal than IRCR or WFT, but it may be useful to some people.

# Windows Forensic Toolchest™ (WFT)



Windows Forensic Toolchest™ (WFT)

The Windows Forensic Toolchest™ (WFT) was written to provide an automated incident response [or even an audit] on a Windowssystem while collecting security-relevant information from thesystem. WFT is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner.

System Information

System Name: MONTY-LAPTOP
Operating System: Microsoft Windows XP Worl
User Name: mmcdoug
Windows Directory: C:\WINDOWS
System Directory: C:\WINDOWS\System32
System Date/Time: 06/01/2007 15:10:53 (24h)

Case Information

Investigator Name: Monty
Case ID: Test

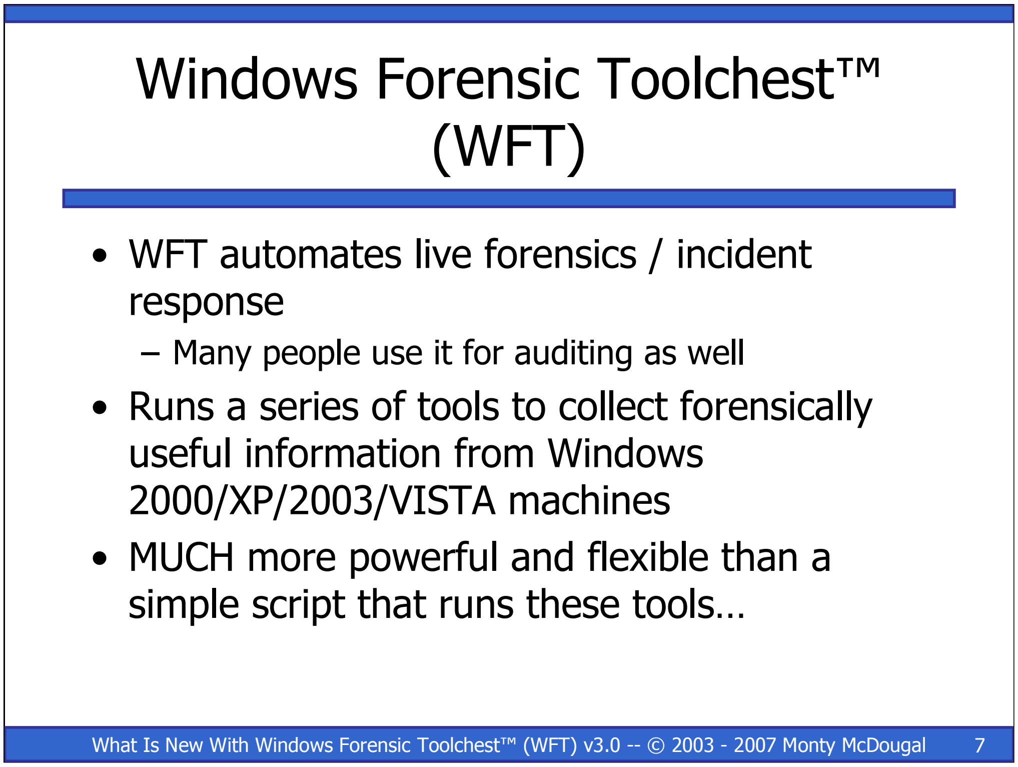
System: MONTY-LAPTOP (mmcdoug) (Monty)

Windows Forensic Toolchest™ (WFT) v3.0.01

S:\MyCVSwork\Programs\wft3\code\Release\wft.exe

```
=============================================================================
Windows
Forensic
Toolchest TM
            <WFT>                                  v3.0.01
=============================================================================
Copyright (C) 2003-2007 Monty McDougal.  All rights reserved.
http://www.foolmoon.net/security/
=============================================================================

                    NON-COMMERCIAL USE ONLY
```

Basic screenshot of the primary interfaces a WFT v3.0 user interfaces with. The DOS collection window and the HTML output reports.



# Windows Forensic Toolchest™ (WFT)

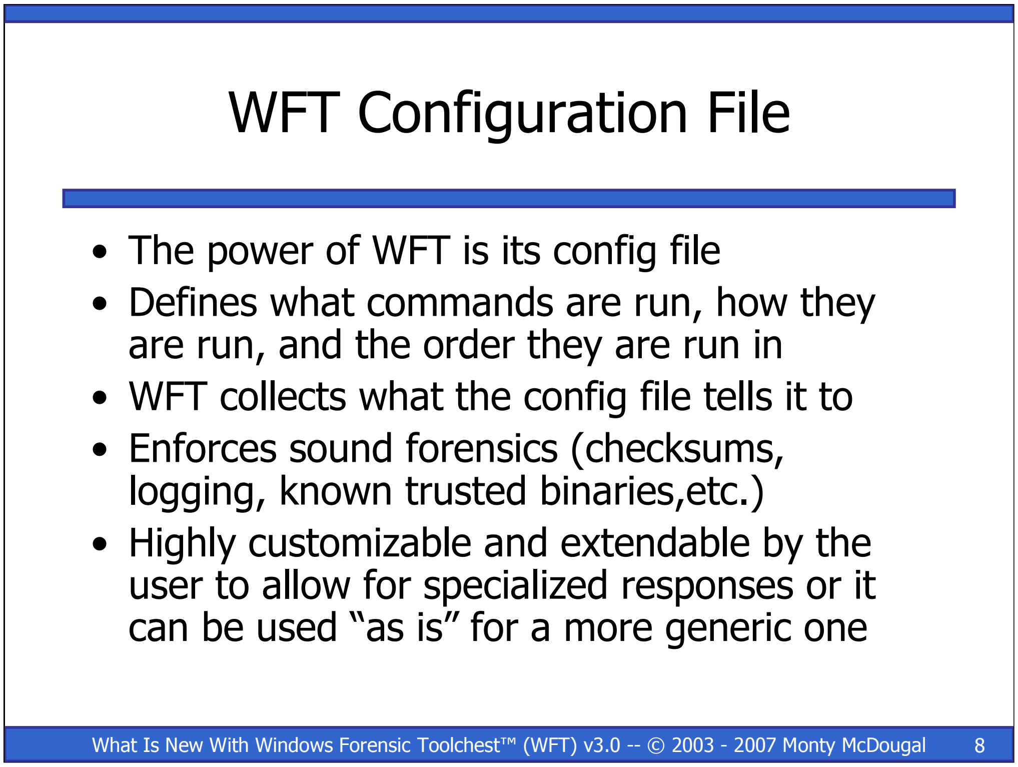
- WFT automates live forensics / incident response
  - Many people use it for auditing as well
- Runs a series of tools to collect forensically useful information from Windows 2000/XP/2003/VISTA machines
- MUCH more powerful and flexible than a simple script that runs these tools…

The Windows Forensic Toolchest™ (WFT) was written to provide an automated incident response [or even an audit] on a Windows system and collect security-relevant information from the system. It is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner. A knowledgeable security person can use it to help look for signs of an incident (when used in conjunction with the appropriate tools). WFT is designed to produce output that is useful to the user, but is also appropriate for use in court proceedings. It provides extensive logging of all its actions along with computing the MD5/SHA1 checksums along the way to ensure that its output is verifiable. The primary benefit of using WFT to perform incident responses is that it provides a simplified way of scripting such responses using a sound methodology for data collection.

The author of this tool is open for suggestions, criticisms of this tool, or offers to help improve the tool’s config file and/or its documentation. Comments relating to WFT can be sent to the author at wft@foolmoon.net.

WFT and the GCFA practical paper which discuss it are available from:

**http://www.foolmoon.net/security**



# WFT Configuration File

- The power of WFT is its config file
- Defines what commands are run, how they are run, and the order they are run in
- WFT collects what the config file tells it to
- Enforces sound forensics (checksums, logging, known trusted binaries,etc.)
- Highly customizable and extendable by the user to allow for specialized responses or it can be used “as is” for a more generic one

This is the config file format used by **WFT 3.0**:

**ACTION EXECUTABLE WFTCHECKSUM COMMAND OUTPUT MENU DESCRIPTION**

Note: Each of these items is separated by a TAB (white space will not work).

Note: Lines beginning with # are treated as comments.

**ACTION** tells Windows Forensic Toolchest™ (WFT) how to process each line:

**V** Perform MD5 verification of EXECUTABLE.

**E** Build a COMMAND to execute.

**N** COMMAND produces NO output to md5.

**H** Build a HTML report.

**M** Add a menu heading.

**S** Skip COMMAND if -noslow option is used.

**W** Skip COMMAND if -nowrite option is used.

**WFT 3.0** adds new Valid **ACTIONS**:

**I** Outputs "NOTE:" information from DESCRIPTION before execution

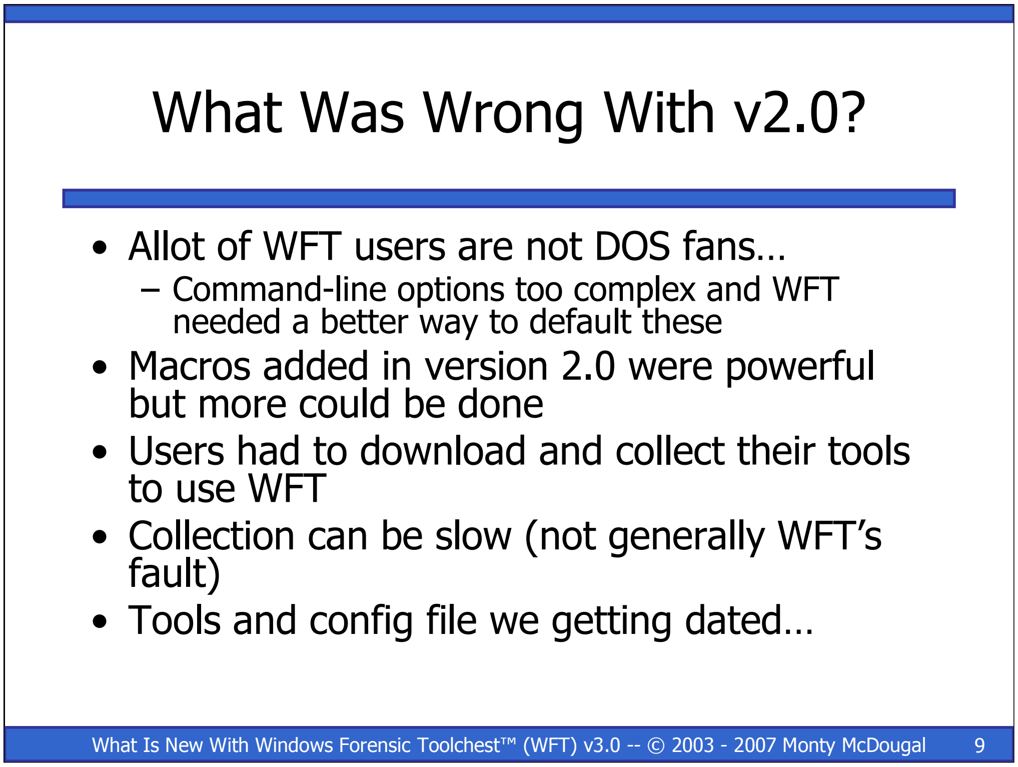
**O** Only executes if EXECUTABLE prior to '\' matches OS. Also required for OS file hash lookups

**P** Prompt before running command if -prompt option is used

**T** Add a tool list report entry

**R** Hacks for tools with known issues such as sysinternals new /accepteula

Note: Multiple ACTIONS can be combined on a line



## What Was Wrong With v2.0?

- Allot of WFT users are not DOS fans…
  - Command-line options too complex and WFT needed a better way to default these
- Macros added in version 2.0 were powerful but more could be done
- Users had to download and collect their tools to use WFT
- Collection can be slow (not generally WFT’s fault)
- Tools and config file we getting dated…

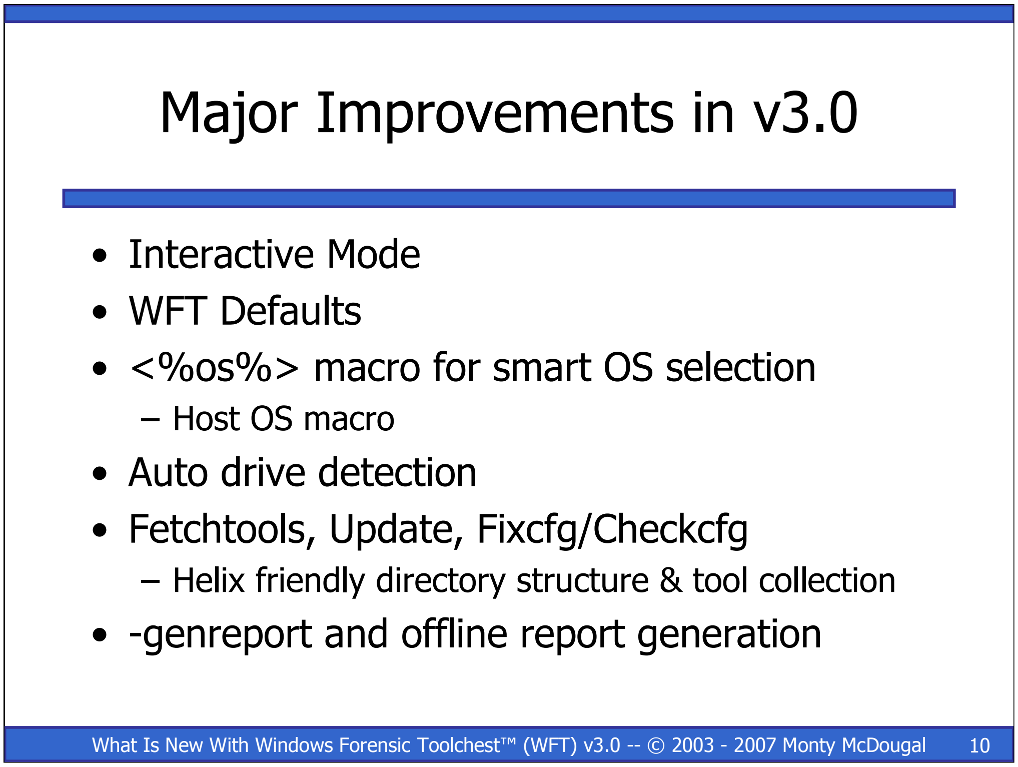
DOS has continued to elude a number of people who are otherwise interested in having capabilities offered by WFT.  For god or bad, we live in a point-and-click world and WFT’s DOS command-line option are a large source of questions and problems for many WFT users.

WFT v2.00 added a very powerful “macro” capability to the config file.  This allowed for dynamic expansion of several options provided by the user at run time.  In some cases, these could have been defaulted based on analyzing the running system.  Why should the user be forced to specify all drive letters when the tool can figure that out for itself?

I have received many, many comments from users stating WFT is great but why don’t you distribute the tools it calls.  Unfortunately, several of the tools are OS based or third-part copyrighted tools with very restrictive licenses.  Being as I don’t want to get sued, I am not going to distribute something I have not been permission to do so.  This problem is further complicated by the fact that some of these tools come and go based over time.  Several of these tools have a useful purpose but are no longer publicly available.  There has to be a better way…

WFT data collection can take a very long time.  This is generally a function of the time it takes to retrieve information from the system.  User could always use –noslow or edit their config files to limit the time required to collect data.  WFT data collection could however be accelerated by moving report generation into an offline processing task.

WFT’s tools and config file were getting updated.  The author was busy (lazy?) and had other priorities given the WFT 2.0 “donation” model had failed so miserably… there had to be a way to address this…



## Major Improvements in v3.0

- Interactive Mode
- WFT Defaults
- <%os%> macro for smart OS selection
  - Host OS macro
- Auto drive detection
- Fetchtools, Update, Fixcfg/Checkcfg
  - Helix friendly directory structure & tool collection
- -genreport and offline report generation

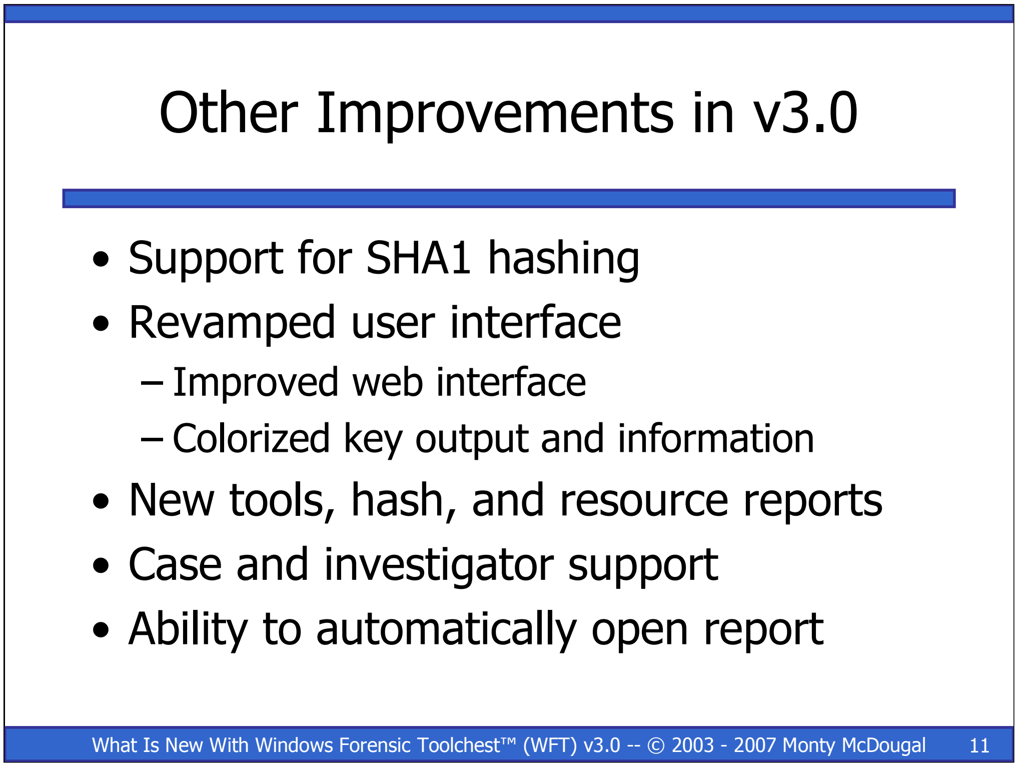
Interactive Mode addresses the limitation of DOS for many users. It provides DOS based prompting for relevant WFT arguments rather than providing these via the command line. This is now the default if WFT is invoked with no arguments unless overridden.

WFT defaults provides a way to override both the interactive defaults but also the defaults of WFT when run on the command line.

New <%os%> macros provide capability to run WFT against multiple OSes while having a single config file. WFT can also use many of the host OS tools if desired (best if only used for auditing purposes). Drive macros and OS can be defaulted at run-time based on the system WFT is being run on.

Fetchtools and update provide an automated way for both updating WFT but also downloading the tools invoked by WFT. Fixcfg and checkcfg have been updated and enhanced to address repairing WFT config files after tools are updated. WFT moved to a directory structure model for tools that is similar to what Helix was using to make Helix and WFT more uniform and to aid in tool acquisition.

Genreport was written to allow reports to be generated post-collection which saves time in the field when collecting data.



## Other Improvements in v3.0

- Support for SHA1 hashing
- Revamped user interface
  - Improved web interface
  - Colorized key output and information
- New tools, hash, and resource reports
- Case and investigator support
- Ability to automatically open report

What Is New With Windows Forensic Toolchest™ (WFT) v3.0 -- © 2003 - 2007 Monty McDougal 11

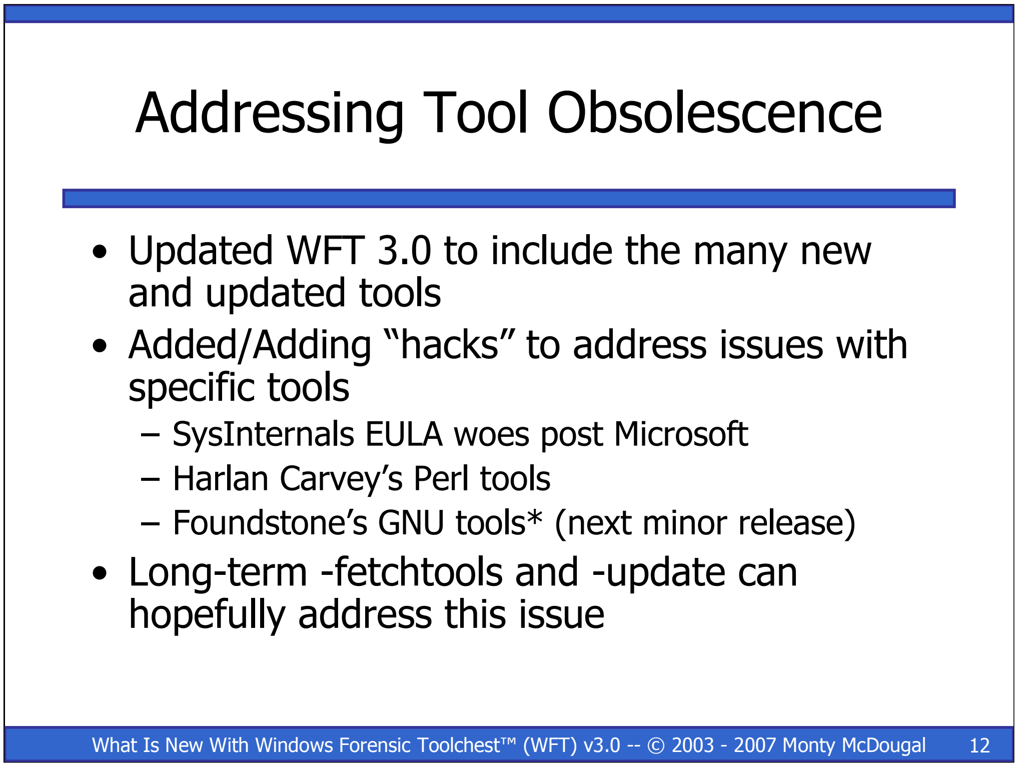
SHA1 hashing was added as an alternative to MD5 hashing.

User interfaces have been improved to include colorization of the output and new web interfaces. Colors can be customized via the WFT defaults file.

Added support for new reports including a comprehensive tool list, list of hash outputs for all files, and updates security resource links

WFT now has the ability to specify and report on case and investigator during execution for collection.

WFT will now automatically open final report in web browser when complete (if desired).



## Addressing Tool Obsolescence

- Updated WFT 3.0 to include the many new and updated tools
- Added/Adding "hacks" to address issues with specific tools
  - – SysInternals EULA woes post Microsoft
  - – Harlan Carvey's Perl tools
  - – Foundstone's GNU tools* (next minor release)
- Long-term -fetchtools and -update can hopefully address this issue

What Is New With Windows Forensic Toolchest™ (WFT) v3.0 -- © 2003 - 2007 Monty McDougal 12

There were several new tools that have been created which would be useful to WFT but were not in the default WFT 2.0 config file. Many of these have been added to the new WFT 3.0 config file.

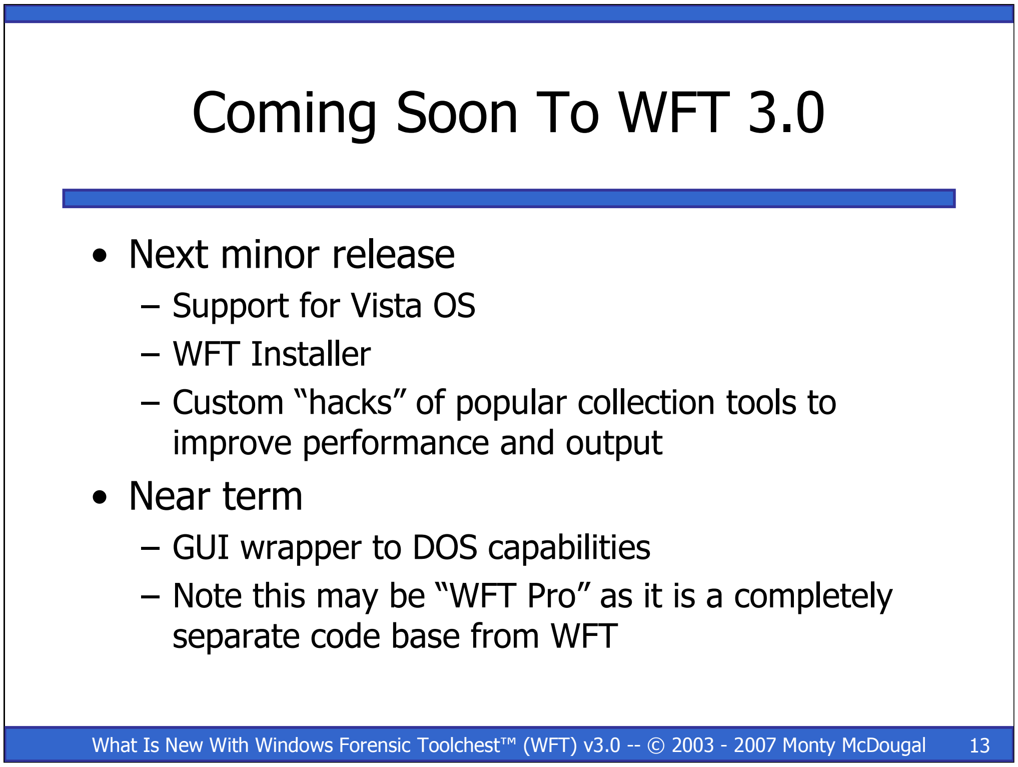
Several of the tools that have been updated or included in WFT had minor issues which could be corrected.

SysInternals new EULA makes running their tools on new systems without human interaction "problematic". Some of these tools support a –accepteula option which address this, but it is inconsistently implemented across their tools. Note WFT 3.0 "fixes" this limitation by making the appropriate registry modifications to allow these tools to run as they did before. Understand this making a minor but necessary modification to the system if you want to be able to collect this valuable data.

Harlan Carvey has implemented several useful tools written in Perl but compiled with perl2.exe. The original binaries from these were compiled with an older version of perl2.exe which had a nasty habit of writing temp files out to disk. Harlan and I worked together to get new version sof these built from the old source code I had for these tools. Unfortunately, there are some issues with mac.exe which require it being updated again and I have reverted to the original version of that utility.

Foundstone also provides utilities I find useful as part of their Forensic Toolkit™. Unfortunately, sfind and hfind were designed for being run interactively via the console and are not ideal for being redirected. This will be tweaked in a future derived release of these tools provided in the next WFT 3.0 minor release.

Finally, I have attempted to address both WFT updates and tool updates via the new –fetchtools and –update options. I expect these to be high-maintenance options. I am not confident I have the right solution to this problem yet, but we will see how this works out and make changes if it does not.



# Coming Soon To WFT 3.0

- Next minor release
  - Support for Vista OS
  - WFT Installer
  - Custom “hacks” of popular collection tools to improve performance and output
- Near term
  - GUI wrapper to DOS capabilities
  - Note this may be “WFT Pro” as it is a completely separate code base from WFT

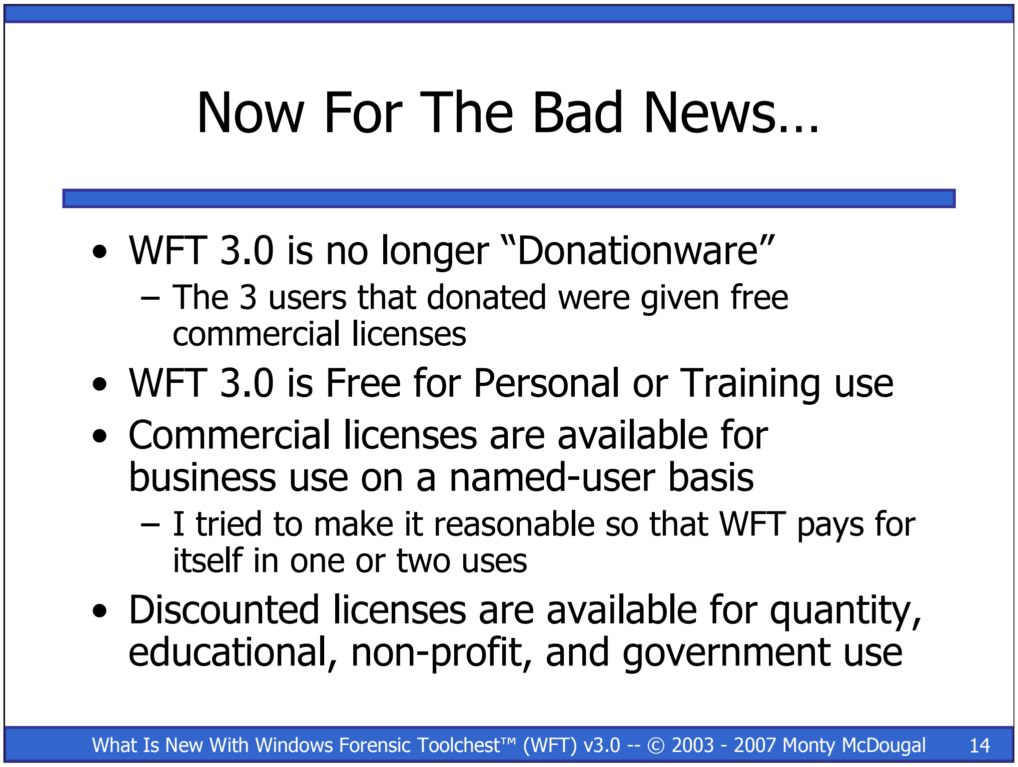
The next minor WFT release should be out in the next 2-3 weeks.

The primary enhancement in this version is support for Vista. It should be noted that Vista’s internal protections by default limit the capability of some tools. The most major loss is DD. There are commercial versions of this which work. I have a couple ideas how to address this as well when I have time…

The next version will also come with an installer. Does it need an installer? No, but it does make it easy for people that don’t like DOS. Since you really need to install WFT on a initial machine to do tool collection, I see no reason not to have an installer.

I am also tweaking a few tools to include with WFT as described on previous slide.

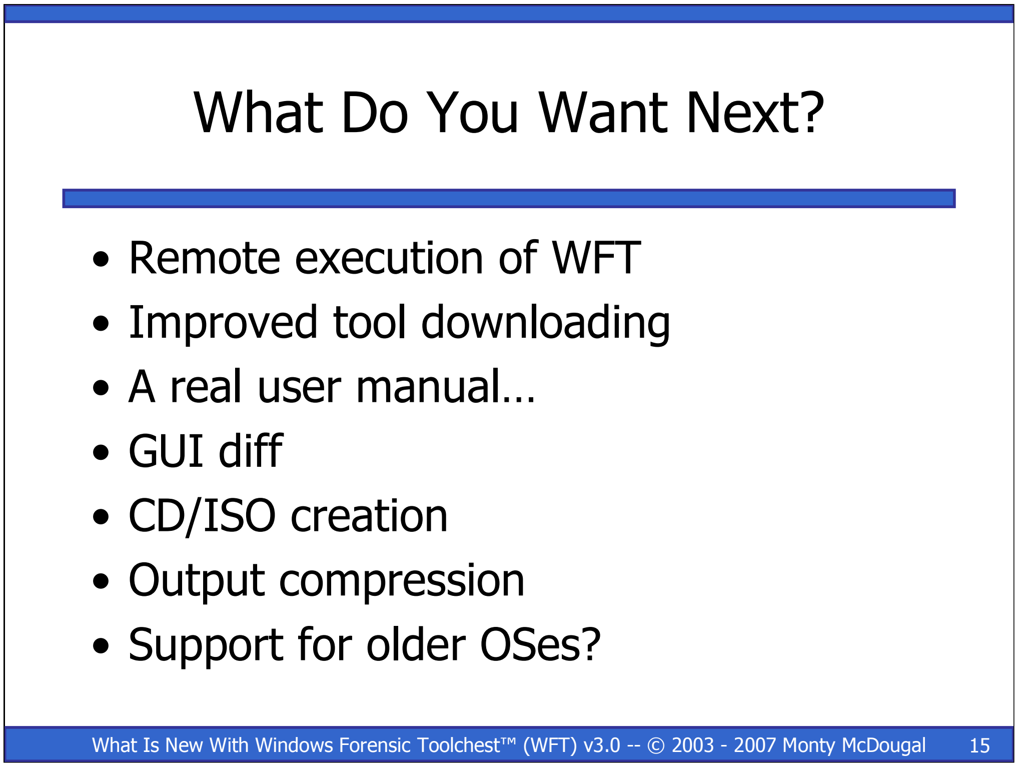
Slightly longer term, I have the new GUI version of WFT in Beta. I will demo it here today. It will be released when I am satisfied it is stable but it is holding up well in initial testing.



# Now For The Bad News…

- WFT 3.0 is no longer “Donationware”
  - The 3 users that donated were given free commercial licenses
- WFT 3.0 is Free for Personal or Training use
- Commercial licenses are available for business use on a named-user basis
  - I tried to make it reasonable so that WFT pays for itself in one or two uses
- Discounted licenses are available for quantity, educational, non-profit, and government use

OK, now for the bad news. After 4 years, WFT 3.0 is no longer free. I tried to make it cheap. I don’t want money stopping people from using WFT but I would like it to support itself. WFT has been widely used by companies that were specifically using WFT to make money, including several companies using it well outside its original license. I have, to date, chosen not to make issue of this. Version 3.0 has a new license geared specifically at stopping this abuse.

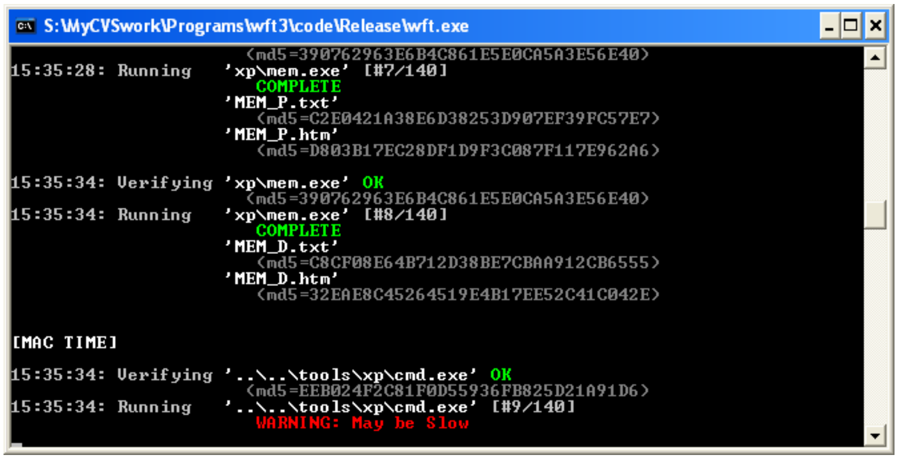


## What Do You Want Next?

- Remote execution of WFT
- Improved tool downloading
- A real user manual…
- GUI diff
- CD/ISO creation
- Output compression
- Support for older OSes?

These are a few of the things I plan on adding in future WFT version. I am interested in feedback on what users would find most beneficial to prioritize the development. I am also interested in ideas I may not have thought of which would help WFT users in their jobs.

# WFT In Action

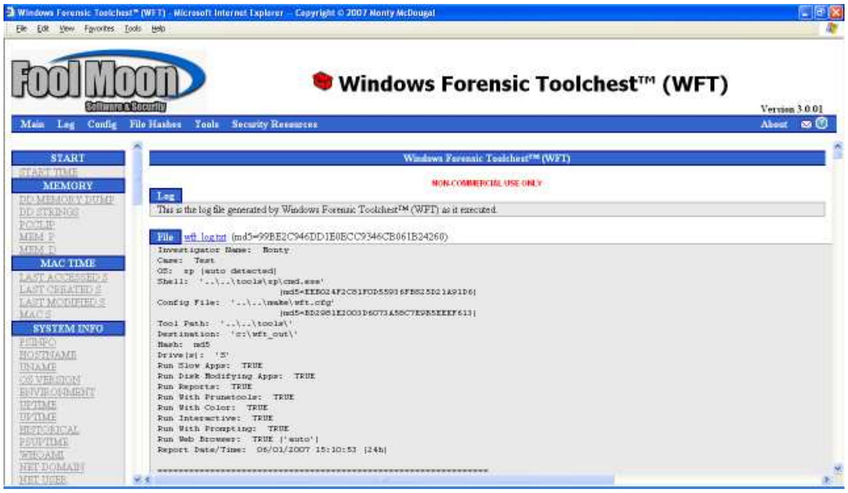


S:\MyCVSwork\Programs\wft3\code\Release\wft.exe

```
                     <md5=390762963E6B4C861E5E0CA5A3E56E40>
15:35:28: Running   'xp\mem.exe' [#7/140]
                      COMPLETE
                    'MEM_P.txt'
                      <md5=C2E0421A38E6D38253D907EF39FC57E7>
                    'MEM_P.htm'
                      <md5=D803B17EC28DF1D9F3C087F117E962A6>

15:35:34: Verifying 'xp\mem.exe'  OK
                     <md5=390762963E6B4C861E5E0CA5A3E56E40>
15:35:34: Running   'xp\mem.exe' [#8/140]
                      COMPLETE
                    'MEM_D.txt'
                      <md5=C8CF08E64B712D38BE7CBAA912CB6555>
                    'MEM_D.htm'
                      <md5=32EAE8C45264519E4B17EE52C41C042E>


[MAC TIME]

15:35:34: Verifying '..\..\tools\xp\cmd.exe'  OK
                     <md5=EEB024F2C81F0D55936FB825D21A91D6>
15:35:34: Running   '..\..\tools\xp\cmd.exe' [#9/140]
                      WARNING: May be Slow
```

This shows a capture of Windows Forensic Toolchest™ (WFT) in action.  In this case, the output. Version 3.0 as it shows one of the most useful new macros added in v3.0 the %os% macro was expanded to “xp” at run-time by WFT.

# Example WFT Reports



This is the log file generated by Windows Forensic Toolchest™ (WFT) as it executed.

wft_log.txt (md5=99BE2C946DD1E0ECC9346CB061B24260)

```
Investigator Name:  Monty
Case:  Test
OS:  xp (auto detected)
Shell:  '..\..\tools\xp\cmd.exe'
                         (md5=EEB024F2C81F0D55936FB825D21A91D6)
Config File:  '..\..\make\wft.cfg'
                         (md5=BD2981E2003D6073A58C7E9B5EEEF613)
Tool Path:  '..\..\tools\'
Destination:  'c:\wft_out\'
Hash:  md5
Drive(s):  'S'
Run Slow Apps:  TRUE
Run Disk Modifying Apps:  TRUE
Run Reports:  TRUE
Run With Prunetools:  TRUE
Run With Color:  TRUE
Run Interactive:  TRUE
Run With Prompting:  TRUE
Run Web Browser:  TRUE ('auto')
Report Date/Time:  06/01/2007 15:10:53 (24h)

=================================================================
```

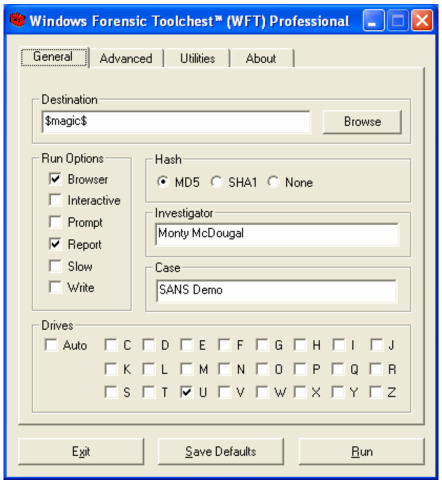
Windows Forensic Toolchest™ (WFT) provides output in two data formats:

HTML Output:  Opening the index.htm file produced by WFT provides an easy to read and easy to navigate interface to the output of the various tools invoked via WFT.  Each of the reports produced under WFT includes the MD5/SHA1 checksum for the binary being run, the exact command line issued to generate the output, a description of the tool, and the output produced by the tool along with the MD5/SHA1 checksum associated with the output.  The HTML reports are designed to be self-documenting via the text provided in the configuration file.

Raw Text Output:  This format allows the viewer to see the output of the individual command exactly as it was produced.  It is generally a bad idea to, in any way, manipulate data being used as evidence in a court of law.  WFT seeks to preserve the original data while providing a user-friendlier HTML version for viewing.  The MD5 checksums produced for each of the output files during collection provides a safeguard to ensure the output can be verified at a later date.

WFT Version 2.0 added two subdirectories for this output – “html” and “txt”.  Additionally it supports system/date/time paths with auto directory creation to better facilitate historical comparisons between WFT runs or systems.  WFT 3.0 adds a new “img” directory.  One of the most important improvements however to reporting is the ability to defer report generation during collection and generate these same reports later in offline mode.

# GUI Interface (BETA)



Destination: $magic$

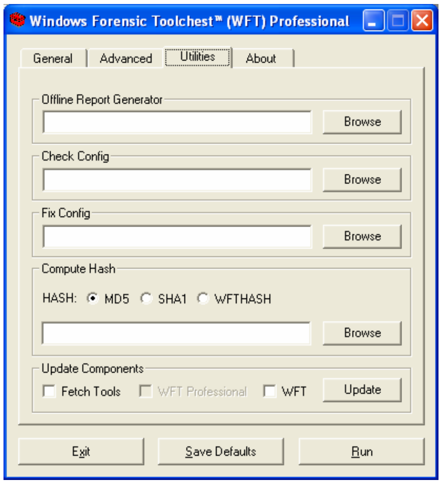
Run Options: ☑ Browser □ Interactive □ Prompt ☑ Report □ Slow □ Write

Hash: ● MD5 ○ SHA1 ○ None

Investigator: Monty McDougal

Case: SANS Demo

Drives: □ Auto □ C □ D □ E □ F □ G □ H □ I □ J □ K □ L □ M □ N □ O □ P □ Q □ R □ S □ T ☑ U □ V □ W □ X □ Y □ Z



Offline Report Generator:

Check Config:

Fix Config:

Compute Hash: HASH: ● MD5 ○ SHA1 ○ WFTHASH

Update Components: □ Fetch Tools □ WFT Professional □ WFT

These are screen captures from the in-work version of the new WFT GUI. Note it supports all the capabilities of the WFT command-line. Eventually it will include additional capabilities not available via the command line (CD/ISO generation, Diff reporting, etc.).

# Thank You For Attending

Questions?
Contact Monty McDougal
monty@foolmoon.net

The author of this tool is open for suggestions, criticisms of this tool, or offers to help improve the tool’s config file and/or its documentation. Comments relating to WFT can be sent to the author at wft@foolmoon.net.

About the author: Monty holds the following major degrees and certifications: BBA in Computer Science / Management (double major) from Angelo State University, MS in Network Security from Capitol College, CISSP, ISSEP, ISSAP, GIAC Certified Incident Handler (GCIH), GIAC Certified Forensic Analyst (GCFA), GIAC Certified UNIX Security Administrator (GCUX), GIAC Certified Windows Security Administrator (GCWN), GIAC Auditing Wireless Networks Certificate (GCAW-C) and serves on the SANS Advisory Board.