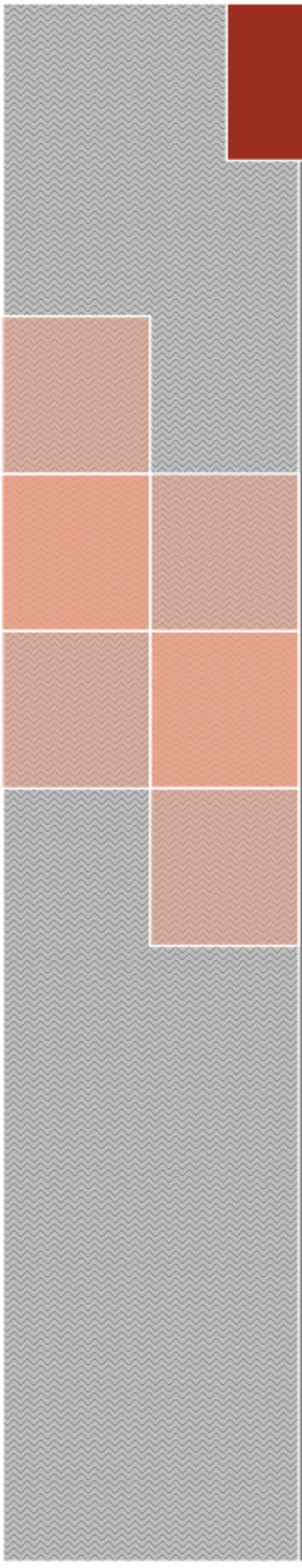

Fulcrum

User's Manual for Version 0.6.1

[redacted]

11/15/2011

1 TABLE OF CONTENTS

2 INTRODUCTION

Fulcrum is a pro-active capability which facilitates the use of a controlled machine to pivot to another uncompromised target machine that is on the same remote LAN. The application will perform a man-in-the-middle attack on the target computer. The application will then monitor the target machine's HTTP traffic and redirect the target to the provided URL when the proper conditions are met.

To be clear, Fulcrum is not is an exploit or a worm. It will not gain arbitrary code execution on a remote machine nor will it perform privilege escalation on the pivot machine. It will not crash applications or operating systems on the pivot or target machines. Fulcrum will not replicate itself or automatically target machines on a LAN nor will it work across a router boundary.

Simply put, Fulcrum will direct a target machine's HTTP client traffic to the URL of the attacker's choice.

2.1 PURPOSE

This is the User's Manual for the initial production release, Version 0.6, of the Fulcrum product. The purpose of this document is to guide end-users on all technical manners surrounding the proper use of the Fulcrum product. This guide includes step-by-step tutorials, information on supported environments, reference information, and known issues.

2.2 INTENDED AUDIENCE

This document is intended primarily for the end users of the Fulcrum product and to a lesser extent the testers and developers.

2.3 TERMINOLOGY

- **Pivot Machine** – The machine where Fulcrum will run.
- **Target Machine** – The machine that Fulcrum will target with its man-in-the-middle and HTTP traffic injection capabilities.
- **Deployment Preparation Machine** – The machine where Fulcrum is prepared and configured for deployment.

2.4 PRODUCT COMPONENTS

The product consists of three separate binaries: FULCRUM, FULCRUMSHUTDOWN, and FULCRUMENCRYPTER.

The FULCRUM binary is the primary application of the product. It is deployed to the **Pivot Machine** and is responsible for performing the actual pivoting technique.

FULCRUMSHUTDOWN is a helper utility which can be deployed to the P**ivot Machine** in order to explicitly initiate a shutdown of the FULCRUM application.

FULCRUMENCRYPTER is a helper utility used on the **Deployment Preparation Machine** to manipulate Fulcrum's configuration and log files.

Four high-level objectives were identified and prioritized for this project. In order of highest to lowest priority, they are:

1. **Correctness** – Correct Target, Correct Network, Successful Injection
2. **Stability** – Don't crash the system, the application, or the process.
3. **Stealth** – Remain imperceptible to the user, avoid Personal Security Product (PSP) Detection, avoid Intrusion Detection System (IDS) detection, and don't get caught.
4. **Usability** – Avoid human errors (easy to configure, easy to deploy), Manage Application Size (large binaries present a problem), Manage Resource Usage

2.5 ANATOMY OF THE PIVOT

There are two basic components to this pivoting technique: the ARP based man-in-the-middle (MITM) and TCP session hijack for HTTP traffic injection. Specially crafted HTTP responses are sent to the target in response to HTTP requests made by the target by hijacking the TCP session. These responses deliver the originally requested content as well as the wax.

2.5.1 ARP SPOOFING TO GET IN THE MIDDLE

The Address Resolution Protocol (ARP) is the network protocol used to resolve OSI Layer 3 Network Addresses (e.g. IPv4 addresses) into OSI Layer 2 Link Addresses (e.g. MAC address). Although ARP has been implemented for a number of combinations of Layer 3 and Layer 2 implementations, Fulcrum is focused only on the Internet Protocol Version 4 (IPv4) and IEEE 802.3 (Ethernet) environment. The combination of IPv4 and Ethernet represents the overwhelming majority of Local Area Networks (LAN). When a computer wants to send data to another computer on an Ethernet network, it must first translate the IP address of the remote machine into its corresponding MAC address. This information is then used to form an Ethernet Frame containing, among other things, the IP packet and the data payload. In a switched Ethernet environment (which is the most common), the MAC address information in the Ethernet Frame is then used to route the frame from the requesting machine to the remote machine. As a result peer machines on a LAN do not see the vast majority of traffic that is generated by each other.

ARP Spoofing is a technique used on a LAN to allow an attacker's machine to intercept data frames from peer machines that were intended for other destinations. This places the attacker's machine in the middle of any traffic from the target's machine to any other destination and is known more commonly as the man-in-the-middle. ARP Spoofing compromises the targets machine's translation of IPv4 addresses into MAC addresses by sending spoofed ARP packets which associate the attacker's MAC address with IP address of another host (such as the default gateway). Any traffic meant for that IP

address would be mistakenly sent to the attacker instead.[1] Refer to Figure 1 - ARP Spoof to visualize the technique.

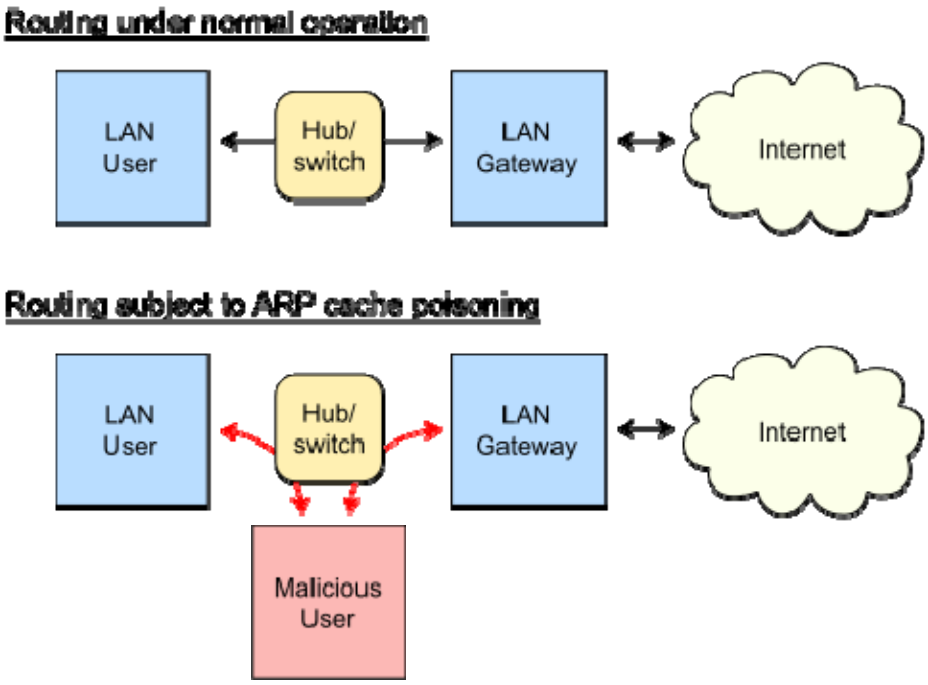

FIGURE 1 - ARP SPOOF

Fulcrum uses ARP spoofing to get in the middle of the target machine and the default gateway on the LAN so that it can monitor all traffic *leaving the target machine*. It is important to note that Fulcrum only establishes itself in the middle on one side of the two-way communication channel between the target machine and the default gateway. Once Fulcrum is in the middle, it forwards all requests from the target machine to the real gateway.

2.5.2 HTTP TRAFFIC INJECTION

Once all network traffic from the target machine is routed to the pivot machine, Fulcrum monitors for specific HyperText Transport Protocol (HTTP) messages. Fulcrum waits for an opportunity to arise to direct the target's HTTP client to retrieve and render content controlled by the pivot machine. When the condition occurs (such as an HTTP GET request), Fulcrum responds by sending a specially formed HTTP packet to the target machine. It is important to note again that Fulcrum is forwarding all traffic from the target machine to the real gateway and is only in the middle of one side of the conversation. As a result, Fulcrum's specially crafted packet must beat the response packet from the real destination (e.g. www.somedomain.com). If the injected packet arrives after the real response, the target machine will simply discard it and the HTTP client will not receive or render it.

[1] http://en.wikipedia.org/wiki/ARP_spoofing

3 SUPPORTED ENVIRONMENTS

NOTE: Although the applications may run properly in operating environments beyond those listed below, only those listed are required to work and will receive testing.

3.1 OPERATING SYSTEMS

The FULCRUM and FULCRUMSHUTDOWN binaries support the following operating systems:

- Windows XP
  - Editions: Home, Professional
  - Architectures: 32-bit only
  - Service Packs: SP0 (RTM), SP1, SP2, SP3, Latest Patch Level as of Fulcrum Release Date
- Windows Vista
  - Editions: Home Basic, Home Premium, Business, Ultimate
  - Architectures: 32-bit, 64-bit
  - Service Packs:SP0 (RTM), SP1, SP2, Latest Patch Level as of Fulcrum Release Date
- Windows 7
  - Editions: Home Premium, Professional, Ultimate
  - Architectures: 32-bit, 64-bit
  - Service Packs: SP0 (RTM), SP1, Latest Patch Level as of Fulcrum Release Date

The FULCRUMENCRYPTER binary supports the following operating systems:

- Windows XP
  - Editions: Professional
  - Architectures: 32-bit only
  - Service Packs: SP3 w/ Latest Patch Level as of Fulcrum Release Date
- Windows Vista
  - Editions: Ultimate
  - Architectures: 64-bit
  - Service Packs: SP3 w/ Latest Patch Level as of Fulcrum Release Date
- Windows 7
  - Editions: Professional, Enterprise
  - Architecture: 64-bit
  - Service Packs: SP1 w/ Latest Patch Level as of Fulcrum Release Date

3.2 HARDWARE

The FULCRUM, FULCRUMSHUTDOWN, and FULCRUMENCRYPTER applications run on any reasonably modern x86-compatible hardware subject to the minimum requirements of the Supported Operating Systems discussed in Section 3.1. Reasonably modern is defined as

- Processor: Intel x86 compatible, Pentium 4 or newer
- RAM: 256MB total system memory or greater

- Disk: 20GB disk or greater
- Wired Network: 10/100/1000Mbps Ethernet
- Wireless Network: 802.11a/b/g/n (optional)

Alternative architectures such as IA32, IA64, ARM and other embedded system architectures are explicitly not supported.

3.3 NETWORKS

The FULCRUM application only supports pivoting on networks that use IPv4 and Ethernet.

3.4 INTERNATIONALIZATION

The FULCRUM, FULCRUMSHUTDOWN, and FULCRUMENCRYPTER binaries support only the multi-byte character set (MBCS)[2] character encoding, specifically the ASCII character set.

NOTE: Although support for additional character encodings such as UNICODE and character sets[3] such as Windows-1252 (Western Languages), Windows-1251(Cyrillic Alphabets), and Windows-1256(Arabic) may be added in the future, they are not supported or tested in this version.

[2] MSDN: Unicode and MBCS http://msdn.microsoft.com/en-us/library/cwe8bzh0(v=VS.90).aspx
[3] Wikipedia: Character Encoding http://en.wikipedia.org/wiki/Character_encoding

4 VERSION INFO

The specific version of a Fulcrum binary can be obtained by looking at the Version Info in either Windows Explorer (Figure 2) or CFF Explorer (Figure 3).

To view this information using Windows Explorer:

1. In Windows Explorer, **browse** to the location of the Fulcrum DLL (e.g. f32.dll)
2. **Right-click** on the DLL and select **Properties**
3. Click the **Details** tab
4. The version information is located in the **File Version** and **Product Version** fields.

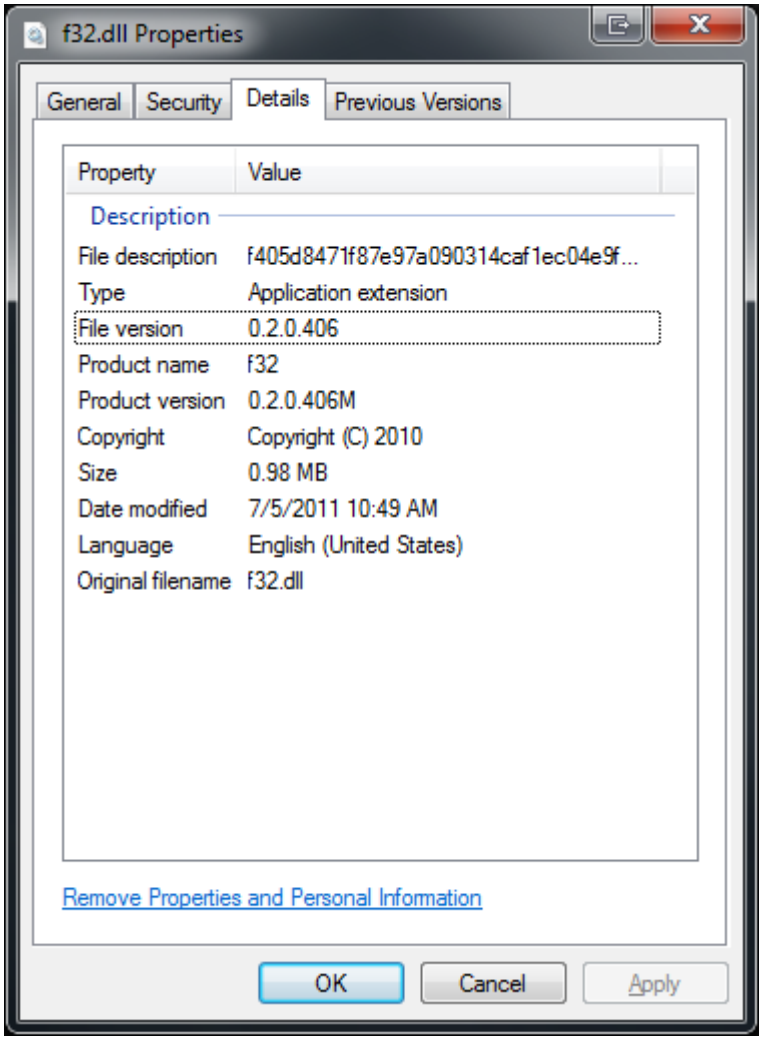

FIGURE 2 - FULCRUM F32.DLL VERSION INFO IN WINDOWS EXPLORER

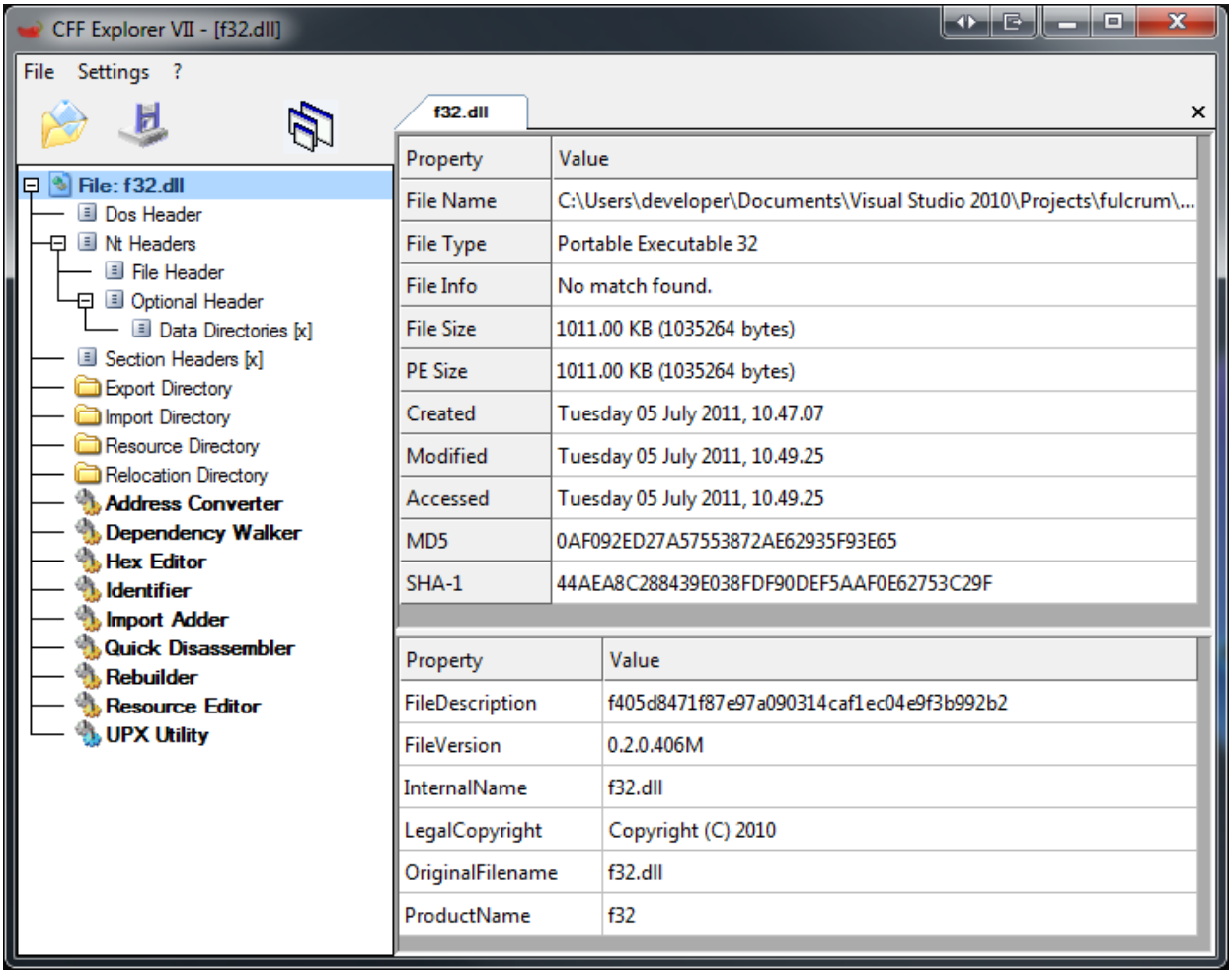

FIGURE 3 - FULCRUM F32.DLL VERSION INFO IN CFF EXPLORER

4.1 LOCAL MODIFICATIONS

Notice in the *Product Version* field in Figure 2 and the *File Version* in Figure 3 that the version number ends with an 'M'. This indicates that this binary was built with local modifications to the source code that have not yet been committed to the version control server or put through the testing and acceptance procedures. **It is therefore important that binaries with an 'M' in their version strings NOT be used in production.**

5 CONFIGURATION

5.1 FULCRUM

There are three ways to provide configuration data to Fulcrum:

1. Command-line parameters
2. Configuration File (f.cfg)
3. Compiled Parameters

Fulcrum searches for configuration data in the specific order above, stopping as soon as one of them is located. All *required* configuration fields must be present in their entirety within the method used. If they are not, then Fulcrum will shutdown. In other words, you cannot provide some parameters via command-line and others via configuration file or any other combination of methods. Any optional field that is not present in the method used will be used the built-in defaults.

5.1.1 COMMAND-LINE PARAMETERS

Fulcrum first looks for the presence of command-line parameters when run as an EXE or via rundll32.exe. If there are any parameters at all, then Fulcrum attempts to fulfill all of the required configuration data from the command-line only. If any required parameter is missing, the application will exit with an error code. No optional parameters can be supplied via the command-line and all of them are fulfilled using the application defaults. The order in which the parameters are provided must be exactly as shown.

The usage of command-line parameters is the following

[Victim MACAddress] [Hijack MACAddress] [Milliseconds between Spoofs] [Injected URL]

For example:

AA:AA:AA:AA:AA:AA BB:BB:BB:BB:BB:BB 1000 http://test.com/cool.jpg

5.1.2 CONFIGURATION FILE

NOTE: The Fulcrum configuration file is encrypted with a 256-bit symmetric key using the AES algorithm. The encryption and decryption of the configuration file is done using the FULCRUMENCRYPTER *utility.*

If no command-line parameters are present, Fulcrum will look for a file name f.cfg residing in the same directory that the Fulcrum binary (f32.exe) is located. If this file is found, Fulcrum attempts to decrypt it and acquire the necessary configuration data from it. If there are problems accessing, decrypting, or parsing this file or if any required parameter is missing, then Fulcrum will exit with an error code.

Optional parameters may be supplied in the configuration file and will override the built-in defaults. Each configuration parameter is supplied in the form

PARAMETER_NAME=<Parameter Value>

Here is an example configuration file in its unencrypted form.

VICTIM_MAC=AA:AA:AA:AA:AA:AA
HIJACKED_MAC=BB:BB:BB:BB:BB:BB
MILLISECONDS_BETWEEN_SPOOFS=1000
INJECTED_URL=http://www.cnn.com

INJECTION_METHOD=DOUBLE_FRAME
USABLE_MEDIA_TYPES=text/html,/**
USER_AGENT_WHITELIST=
USER_AGENT_BLACKLIST=

5.1.3 COMPILED PARAMETERS

If no command-line parameters and no configuration file are present, Fulcrum will use the data that was compiled into the application for its required parameters. This provides another method of executing Fulcrum and avoids the configuration file on disk or the command-line parameters appearing in the Task Manager. This was originally added to support in-memory only deployment and execution. While Fulcrum itself can be deployed and executed in an in-memory only fashion, the WPCAP Pro support DLL will write temporary files to the disk and make changes to the registry. It is important to note however, that this information is stored in plaintext in an un-obfuscated manner in the binary.

In order to change these values, it is necessary to get a developer to re-compile the application. The operational need for the continued support of this feature will be reviewed. Then it may either be removed from future versions or altered to allow for easier editing by the end-users without requiring a developer to be involved.

5.1.4 CONFIGURATION OPTIONS

Parameter Name	Description	Acceptable Values	Default Value
VICTIM_MAC	The MAC address of the Target Machine in the form of XX:XX:XX:XX:XX:XX	00:00:00:00:00:01 – FF:FF:FF:FF:FF:FE inclusive	66:77:88:99:AA:BB
HIJACKED_MAC	The MAC address of the Hijacked Machine (typically the Default gateway) in the form of XX:XX:XX:XX:XX:XX This parameter is also used to verify that the	00:00:00:00:00:01 – FF:FF:FF:FF:FF:FE inclusive	BB:CC:DD:EE:FF:00

	application is running on the correct network.		
MILLISECONDS_BETWEEN_SPOOFS	The number of milliseconds to wait between sending ARP spoof packets.	0 – 2,147,483,647 inclusive	1000
INJECTED_URL	The URL to direct the target machine to.	Any valid URL	http://www.msn.com
INJECTION_METHOD	The method of delivering the target URL inside of an HTTP response.	DOUBLE_FRAME or META_REFRESH	DOUBLE_FRAME
USABLE_MEDIA_TYPES	Accepted media types from the target's HTTP request that we SHOULD consider for injection	Comma separated list. NOTE: NO SPACES, I DON'T TRIM	text/html,*/*
USER_AGENT_WHITELIST	White-listed user agent string tokens.	Comma separated list. NOTE: NO SPACES, I DON'T TRIM	<Blank>
USER_AGENT_BLACKLIST	Black-listed user agent string tokens.	Comma separated list. NOTE: NO SPACES, I DON'T TRIM	<Blank>

5.2 FULCRUM SHUTDOWN

FULCRUMSHUTDOWN does not use any configuration method.

5.3 FULCRUM ENCRYPTER

FULCRUMENCRYPTER only uses the command-line configuration method.

The usage of command-line parameters is the following:

[-d|-e] [input_path] [output_path]

For example, to create an encrypted copy of the file f.cfg.decr into a file called f.cfg:

FulcrumEncrypter32.exe –e f.cfg.decr f.cfg

Or to decrypt the log file named f.log into f.log.decr:

FulcrumEncrypter32.exe –d f.log f.log.decr

6 THE END-TO-END PROCESS

The principal by which Fulcrum works is fairly straight-forward and we've tried to keep the tedium in using the product to a minimum. There are however, a few steps you should follow when using the product. A little homework upfront goes a long way to ensuring the greatest chance of success and avoiding any mistakes (You DO want this to work, right?).

6.1.1 PREPARATION

So you are just itching to use Fulcrum against this target of yours and you're ready to dive in! Hang on there partner. First we need to gather the following information before we can build a deployment package:

1. The MAC address of the LAN-side interface of the gateway
2. The MAC address of the target machine
3. The URL to inject into the HTTP response
4. The Injection method of the HTTP response
5. The character set of the pivot machine
6. Any user agent string whitelist entries
7. Any user agent string blacklist entries
8. Any target content type modifications
9. Whether the pivot machine is a laptop or a desktop
10. The OS version of the pivot machine
11. The bitness of the process Fulcrum will run in
12. The privilege level of the process Fulcrum will run in
13. What PSPs are present on the pivot machine
14. How the Fulcrum files will be delivered to the pivot machine
15. Where the Fulcrum files will be deployed to on the pivot machine's file system
16. When Fulcrum should be delivered to the pivot machine
17. How Fulcrum will be started on the pivot machine
18. When Fulcrum should be started
19. If Fulcrum should be automatically restarted
20. When Fulcrum should be shutdown
21. When Fulcrum should be removed

6.1.2 PACKAGING

OK, so you've collected all the information you needed. Fantastic! Pat yourself on the back, grab a fresh caffeinated beverage of your choice and then let's get down to it.

Ready? Now we need to package up the applications and configuration data. This involves:

1. Choosing what run-time execution mode to use (i.e. EXE, rundll32.exe, or LoadLibrary)

2. Choosing what configuration method to use (i.e. command-line parameters, configuration file, or compiled parameters)
3. Selecting the proper binaries based on the required bitness
4. Cross-checking the P2P test results for potential problems
5. Preparing the binary if compiled parameters are used
6. Preparing the command and cross-checking the optional parameters default values if the command-line parameters are used
7. Editing the configuration file and encrypting it if the configuration file method is used

6.1.3 DELIVERY

Are we there yet?

Almost. The delivery phase encompasses all activities to deliver the Fulcrum packages to the pivot machine and providing the binaries with code execution. The bummer is that delivery is outside of the scope of this product and there are numerous manners in which it could be done so we can't provide any awesome tips or huge checklists here.

6.1.4 MANAGEMENT

If you are reading this then you have successfully delivered the Fulcrum packages and provided the binaries with code execution. Hoorah!

At this stage there is not much to do other than sit back and wait. The release builds of the Fulcrum binaries don't print anything to the console nor do they log any messages, so all we hear are the sound of crickets. There are a few management tasks you may need to do however, including:

1. Restarting Fulcrum if it has stopped (such as due to a reboot on the pivot machine)
2. Updating configuration data on the pivot machine to reflect new information (e.g. a new injection URL or target MAC)
3. Manually initiating a shutdown of Fulcrum if you need it to stop or want to remove it
4. Removing the Fulcrum files from the pivot machine

7 STEP-BY-STEP

7.1.1 PREPARE A CONFIGURATION FILE

1. On a Windows machine, **create** a file called *f.cfg.decr*. You can use a copy of the *f.cfg.example* file that comes with the product as a starting point.
2. **Supply** each of the *required parameter values* described in Section 5.1.4 Configuration Options using the information gathered during the process outlined in Section 6.1.1 Preparation.
3. **Supply** any desired *optional parameter values*
4. **Refer** to Section 7.1.3 Encrypting the Configuration File to encrypt the file for use with the FULCRUM application.

7.1.2 UPDATE A CONFIGURATION FILE

When Fulcrum is run as an EXE with command-line parameters, the parameter values including the injection URL are available in plaintext via simple tools like Task Manager or Process Explorer. Compiling the parameters into the application can be tedious when you want to re-use a pivot machine to target multiple machines. You may then find it useful to use a configuration file so that you can just upload the new file and restart Fulcrum to move to the next target.

1. Follow the Steps in Section 7.1.1 Prepare a Configuration File to prepare the new config
2. Shutdown Fulcrum on the pivot machine
3. Place the new f.cfg file next to the Fulcrum binary, overwriting any existing copy if present
4. Start Fulcrum

7.1.3 ENCRYPTING THE CONFIGURATION FILE

The Fulcrum configuration file must be encrypting using the 256-bit key and the AES algorithm in order for the Fulcrum application to use it. This is easily accomplished using the supplied FULCRUMENCRYPTER utility. This utility will create an encrypted copy of the configuration file.

1. **Copy** the FULCRUMENCRYPTER binary (FulcrumEncrypter32.exe) into the same directory as the configuration file.
2. **Open** a *command prompt*
3. **Change** directories to the location of the *f.cfg.decr* file
4. **Encrypt** the configuration file by typing the following command into the command prompt:

FulcrumEncrypter32.exe –e f.cfg.decr f.cfg

7.1.4 DECRYPTING THE CONFIGURATION FILE

If you need to decrypt the configuration file for reason, such as you want to verify its contents or update its contents and you don't have the original decrypted version handy, you again us the FULCRUMENCRYPTER utility.

1. **Copy** the FULCRUMENCRYPTER binary (FulcrumEncrypter32.exe) into the same directory as the configuration file.
2. **Open** a *command prompt*
3. **Change** directories to the location of the *f.cfg* file
4. **Decrypt** the configuration file by typing the following command into the command prompt:

FulcrumEncrypter32.exe –d f.cfg f.cfg.decr

7.1.5 DECRYPTING THE LOG FILE (DEBUG BUILDS ONLY)

Debug builds will generate encrypted log files that contain useful information for developers and testers. To view the contents of the log, you must first decrypt it using the supplied FULCRUMENCRYPTER utility.

1. **Copy** the FULCRUMENCRYPTER binary (FulcrumEncrypter32.exe) into the same directory as the log file.
2. **Open** a *command prompt*
3. **Change** directories to the location of the *f.log* log file.
4. **Decrypt** the log file by typing the following command into the command prompt:

FulcrumEncrypter32.exe –d f.log f.log.decr

7.1.6 RUNNING FULCRUM AS AN EXE WITH A CONFIGURATION FILE

1. **Prepare** a configuration file as described in Section 7.1.1
2. **Encrypt** a configuration file as described in Section 7.1.3
3. **Copy** the FULCRUM executable binary (f32.exe or f64.exe) into the same directory as the configuration file.
4. **Double-click** the FULCRUM binary

7.1.7 RUNNING FULCRUM AS AN EXE WITH COMMAND-LINE PARAMETERS

1. **Copy** the FULCRUM executable binary (f32.exe or f64.exe) to the desired location.
2. **Open** a *command prompt*
3. **Change** directories to the location of the FULCRUM binary.
4. **Execute** the binary by typing the following command into the command prompt:

f32.exe [Victim MACAddress] [Hijack MACAddress] [Milliseconds between Spoofs] [Injected URL]

For example:

f32.exe AA:AA:AA:AA:AA:AA BB:BB:BB:BB:BB:BB 1000 http://test.com/cool.jpg

7.1.8 RUNNING FULCRUM AS AN EXE WITH COMPILED PARAMETERS

1. **Copy** the FULCRUM executable binary (f32.exe or f64.exe) to the desired location.
2. **Double-click** the FULCRUM binary

7.1.9 RUNNING FULCRUM AS A DLL USING RUNDLL32.EXE WITH A CONFIGURATION FILE

1. **Prepare** a configuration file as described in Section 7.1.1
2. **Encrypt** a configuration file as described in Section 7.1.3
3. **Copy** the FULCRUM DLL binary (f32.dll) into the same directory as the configuration file.
4. **Open** a *command prompt*
5. **Change** directories to the location of the FULCRUM binary.
6. **Execute** the binary by typing the following command into the command prompt:

rundll32.exe f32.dll,rundll_entry

7.1.10 RUNNING FULCRUM AS A DLL USING RUNDLL32.EXE WITH COMMAND-LINE PARAMETERS

1. **Copy** the FULCRUM DLL binary (f32.dll) to the desired location.
2. **Open** a *command prompt*
3. **Change** directories to the location of the FULCRUM binary.
4. **Execute** the binary by typing the following command into the command prompt:

rundll32.exe f32.dll,rundll_entry [Victim MACAddress] [Hijack MACAddress] [Milliseconds between Spoofs] [Injected URL]

For example:

rundll32.exe f32.dll,rundll_entry AA:AA:AA:AA:AA:AA BB:BB:BB:BB:BB:BB 1000 http://test.com/cool.jpg

7.1.11 RUNNING FULCRUM AS A DLL USING LOADLIBRARY WITH A CONFIGURATION FILE

1. **Prepare** a configuration file as described in Section 7.1.1
2. **Encrypt** a configuration file as described in Section 7.1.3
3. **Copy** the FULCRUM DLL binary (f32.dll or f64.dll) to the same directory as the configuration file.
4. From the parent process, **load** the binary using *LoadLibrary*
5. From the parent process, **get** the parameterless export using *GetProcAddress("func")*. This function's signature is: *void func(void)*
6. From the parent process, **call** func

7.1.12 RUNNING FULCRUM AS A DLL USING LOADLIBRARY WITH COMPILED PARAMETERS

1. **Copy** the FULCRUM DLL binary (f32.dll or f64.dll) to desired location.

2. From the parent process, **load** the binary using *LoadLibrary*
3. From the parent process, **get** the parameterless export using *GetProcAddress("func")*. This function's signature is: *void func(void)*
4. From the parent process, **call** func

7.1.13 SHUTTING DOWN FULCRUM WITH FULCRUMSHUTDOWN AS AN EXE

1. **Copy** the FULCRUMSHUTDOWN executable binary (fs32.exe or fs64.exe) to the desired location.
2. **Double-click** the FULCRUM binary Shutting Down Fulcrum with FulcrumShutdown as a DLL with rundll32.exe

Shutting Down Fulcrum with FulcrumShutdown as a DLL with LoadLibrary

7.1.14 SHUTTING DOWN FULCRUM WITH FULCRUMSHUTDOWN AS A DLL USING RUNDLL32.EXE

1. **Copy** the FULCRUMSHUTDOWN DLL binary (fs32.dll or fs64.dll) to the desired location.
2. **Open** a *command prompt*
3. **Change** directories to the location of the FULCRUMSHUTDOWN binary.
4. **Execute** the binary by typing the following command into the command prompt:

rundll32.exe fs32.dll,rundll_entry

7.1.15 SHUTTING DOWN FULCRUM WITH FULCRUMSHUTDOWN AS A DLL USING LOADLIBRARY

1. **Copy** the FULCRUMSHUTDOWN DLL binary (f32.dll or f64.dll) to desired location.
2. From the parent process, **load** the binary using *LoadLibrary*
3. From the parent process, **get** the parameterless export using *GetProcAddress("func")*. This function's signature is: *void func(void)*
4. From the parent process, **call** func

7.1.16 REMOVING FULCRUM

1. **Shutdown** FULCRUM using one of the methods described in Sections 7.1.13, 7.1.14, or 7.1.15
2. Delete all of the following files (not all will be present):
    a. f32.exe
    b. f32.dll
    c. fs32.exe
    d. fs32.dll
    e. f.cfg
    f. f.log

8 PERSONAL SECURITY PRODUCTS

The Fulcrum binaries are tested by the IV&V team according to their latest test plan. To see the results of these tests, refer to the latest IV&V Test Report located on DEVNET for this product release.

9 KNOWN ISSUES

- If the pivot machine is moved to another network while Fulcrum is running, the pivot machine will not be able to connect to the internet or generally use networking services. This is because Fulcrum places a static ARP entry for the default gateway in the pivot machine's ARP table/neighbor's cache when the application starts up. This will be addressed in future versions. For now, the recommended work around is not to deploy Fulcrum on machines that are likely to change networks, such as laptops and netbooks.
- MAC addresses must be specified in the form XX:XX:XX:XX:XX:XX using colons, not dashes. In a future version we will likely accept either.
- FulcrumShutdown only works if it is run as the same user with the same privileges that Fulcrum was started with. If Fulcrum is running as NT-AUTHORITY\SYSTEM for example, a normal user or even an administrator cannot shutdown Fulcrum using FulcrumShutdown. In the case of Fulcrum running as the system account, you can run FulcrumShutdown using Sysinternal's psexec tool as the system account using the –s flag. For example: *psexec –s fs32.exe*
- WinPcap leaks a two handles each time Fulcrum is run – one for a registry HKEY and one for the packet[nt|2k|vista].dll. Even if Fulcrum is run thousands of times in the same process, this won't exhaust the handle address space.
- If Fulcrum is run on a pivot machine which is actually a virtual machine and the host machine is running Linux and VMware, then a notification is displayed on the host system. The notification is a message box that states: "*The virtual machine's operating system has attempted to enable promiscuous mode on adapter Ethernet0. This is not allowed for security reasons."*
- Fulcrum does not measure its success or failure based on wax success. Fulcrum bases its success or failure on whether the target machine requests the injected URL.
- If the target machine goes offline and the pivot machine doesn't notice for an extended period of time OR if the target machine is online but not generating any traffic for an extended period of time, then the switch that the target and pivot are both connected to may begin sending out the ARP spoof packets to all ports on the switch. This is known as "failing open" and is a result of the target machine's MAC address expiring out of the CAM table on the switch. **All other machines on the switch will discard this traffic unless their interface is in promiscuous mode**. Even if the interface is in promiscuous mode, some operating system versions will not update their ARP cache from these packets and thus will not be ARP spoofed. Finally, for those machines that do have their interfaces in promiscuous mode and update their ARP table from these broadcasted unicast ARP spoof packets, **Fulcrum will still not fire on any of them and will simply route their traffic on to the real gateway**.

-

10 CHANGELOG

Version 0.6.1

- Fixed issue where Fulcrum was continuing to send ARP spoofs while it was ARP scanning to find a target that had gone offline. (~~#124~~)
- Fixed issue where Fulcrum was exiting with the SUCCESS code (0) when an error condition had occurred. (~~#123~~)
- Added routing verification to avoid DoSing the target when a firewall is blocking inbound packets. (~~#119~~)

Version 0.6.0

- Fixed issue where WinPcap files were not deleted off of the disk on exit. (~~#103~~)
- Removed load-time dependencies on various DLLs to reduce the IAT fingerprint. (~~#63~~)
- Fixed an issue where you could not shutdown or start new instances of Fulcrum from the same process due to a resource leak in WinPcap. (~~#107~~)
- Fulcrum Shutdown was failing with Exit Code FAILURE_CREATING_SYNCHRO_EVENT when it was run with insufficient privileges. It now returns FAILURE_ACCESS_DENIED. (~~#108~~)
- Updated the signatures of the exported functions to match those required by the calling process. (~~#109~~)
- Fixed an issue with 64-bit builds that caused Fulcrum to fail . (~~#102~~)
- Added the Mission Manager's document.
- Updated user's manual with known issues.
- Updated version numbers to match the finalized versioning scheme for delivery.

Version 0.3.0

- Initial Release for IV&V

11 GLOSSARY OF TERMS

Term	Definition
ARP	Address Resolution Protocol
AV	Anti-Virus
CRT	C-Runtime
DLL	Dynamic-link Library
HTTP	Hyper-text Transfer Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPv4	Internet Protocol Version 4
LAN	Local Area Network
MAC	Media Access Control
MITM	Man-In-The-Middle
PSP	Personal Security Product