# BLACKENERGY & QUEDAGH

## The convergence of crimeware and APT attacks

**TLP: WHITE**

CONTENTS



BlackEnergy is a toolkit that has been used for years by various criminal outfits. In the summer of 2014, we noted that certain samples of BlackEnergy malware began targeting Ukranian government organizations for information harvesting. These samples were identified as being the work of one group, referred to in this document as "**Quedagh**", which has a history of targeting political organizations.

The Quedagh-related customizations to the BlackEnergy malware include support for proxy servers and use of techniques to bypass User Account Control and driver signing features in 64-bit Windows systems. While monitoring BlackEnergy samples, we also uncovered a new variant used by this group. We named this new variant **BlackEnergy 3**.

The use of BlackEnergy for a politically-oriented attack is an intriguing convergence of criminal activity and espionage. As the kit is being used by multiple groups, it provides a greater measure of plausible deniability than is afforded by a custom-made piece of code.


**F-SECURE LABS SECURITY RESPONSE**

Malware Analysis Whitepaper


F-Secure

## INTRODUCTION

BlackEnergy is a popular crimeware (that is, malware designed to automate criminal activities) that is sold in the Russian cyber underground and dates back to as early as 2007. Originally, it was designed as a toolkit for creating botnets for use in conducting Distributed Denial of Service (DDoS) attacks. Over time, the malware has evolved to support different plugins, which are used to extend its capabilities to provide necessary functions, depending on the purpose of an attack.

Given the nature of its toolkit, BlackEnergy has unsurprisingly been used by different gangs for different purposes; some use it for sending spam, others for stealing banking credentials. The most notorious use may be when it was used to conduct cyberattacks against Georgia during the Russo-Georgian confrontation in 2008.

In the summer of 2014, BlackEnergy caught our attention when we noticed that samples of it were now tailored to target Ukrainian government institutions. Though it may be unrelated, it is interesting to note that this change conveniently coincides with the on-going crisis in that country. Related or not, one thing is certain: the actor(s) using these customized BlackEnergy malware are intent on stealing information from the targets. The use of this crimeware in what constitutes as an advance persistent threat (APT) attack is interesting. In 'black operations' (black ops), an important criteria is that the attack should not be attributable - and what provides better plausible deniability than crimeware known to be used by multiple parties?

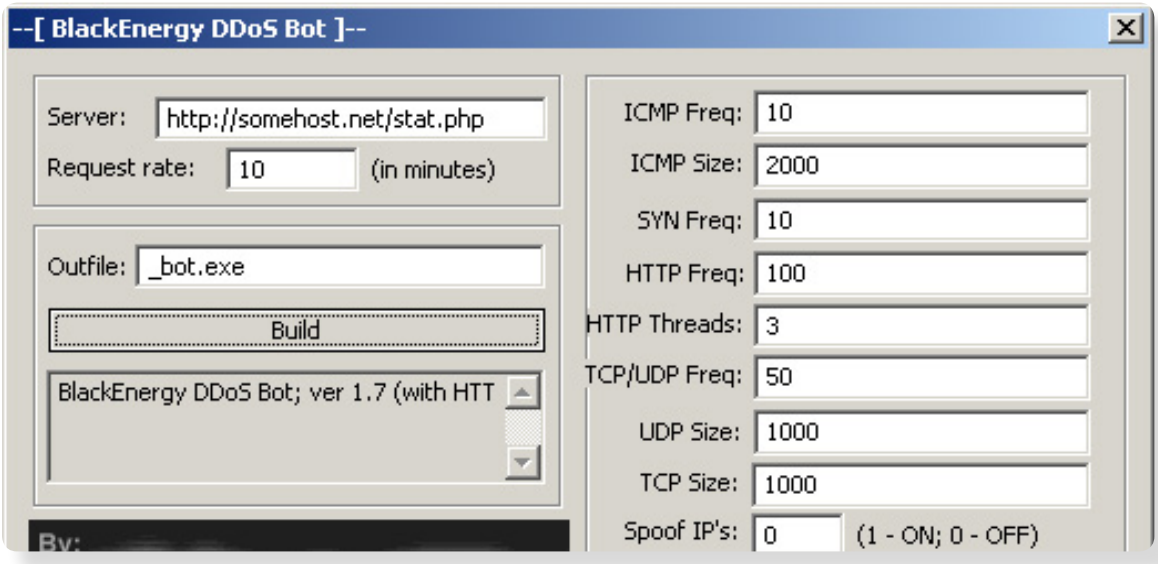
In this paper we focus only on BlackEnergy samples known to be used specifically by the actors we identify as **Quedagh**, who seem to have a particular interest in political targets. Special focus will be on the samples that were used in targeted attacks against Ukrainian government organizations earlier this year.

## ATTACK OVERVIEW

At the time of writing, we have little information on how exactly victims are receiving the BlackEnergy malware being pushed by the Quedagh gang. An educated guess is that they are receiving the malware via targeted emails containing malicious attachments. Meanwhile, the following infection and technical details are based on samples gathered after searching through F-Secure Labs' collection of all BlackEnergy samples and identifying those with Quedagh characteristics.

The BlackEnergy toolkit comes with a builder application which is used to generate the clients that the attacker(s) use to infect victim machines. The toolkit also comes with server-side scripts, which the attackers set up in the command and control (C&C) server. The scripts also provide an interface where an attacker can control his bots. The simplicity and convenience provided by the toolkit means that anyone who has access to the kit can build his own botnet without any skills required.

Figure 1: BlackEnergy Builder from 2007



The original BlackEnergy toolkit first emerged in 2007 and is referred to in this paper as BlackEnergy 1. A later variant of the toolkit (BlackEnergy 2) was released in 2010. We also encountered a previously unseen variant, which had been rewritten and uses a different format for its configuration. It also no longer uses a driver component. We dubbed this new variant **BlackEnergy 3**.

## INFECTION VECTORS

Most of the recent BlackEnergy installers collected are named **msiexec.exe**. We believe they are either dropped by another executable that uses social engineering tricks to mislead the user into executing the installer, or by documents containing exploits that silently perform the installation.

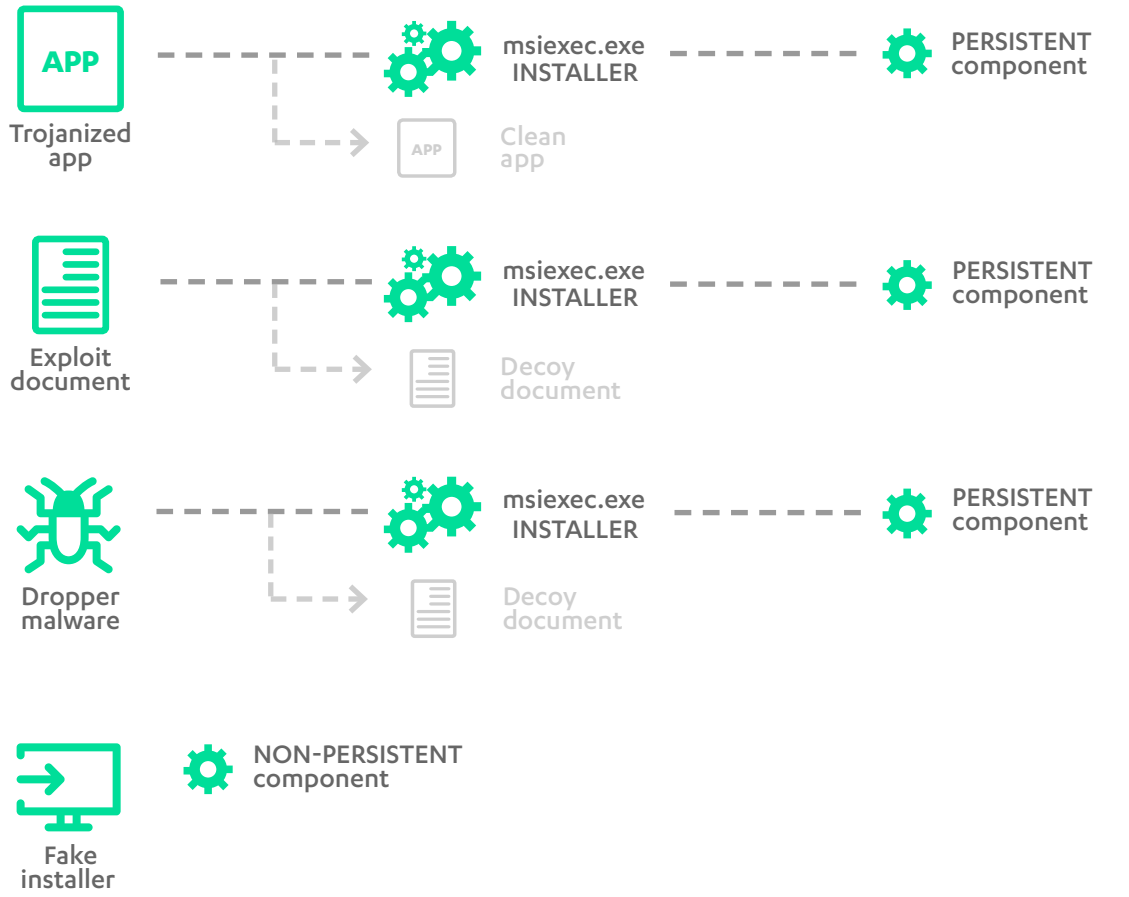
We found at least 2 trojanized legitimate applications that execute the installer (in addition to their legitimate tasks). Trojanization is an effective infection method, as most users have no way of observing that a malicious component is being installed in tandem with a legitimate program.

Some earlier installer variants, then named **regedt32.exe,** were distributed by documents exploiting software vulnerabilities, one of which was CVE-2010-3333. These documents drop and execute the installer, then open a decoy document. It is reasonable to assume that a similar approach has been used to deliver the more recent installer variants.

The installer filename of BlackEnergy 3 is still **msiexec.exe**. However, it is delivered and executed by a dropper which opens a decoy document in the foreground. We also encountered a standalone, non-persistent sample that pretends to be Adobe Flash Player Installer. It does not use any decoy document or application and does not run after reboot.

The overview below summarizes the various infection vectors used by the Quedagh gang to deliver BlackEnergy crimeware to the designated targets.

### OVERVIEW OF INFECTION VECTORS USED AGAINST UKRAINIAN TARGETS

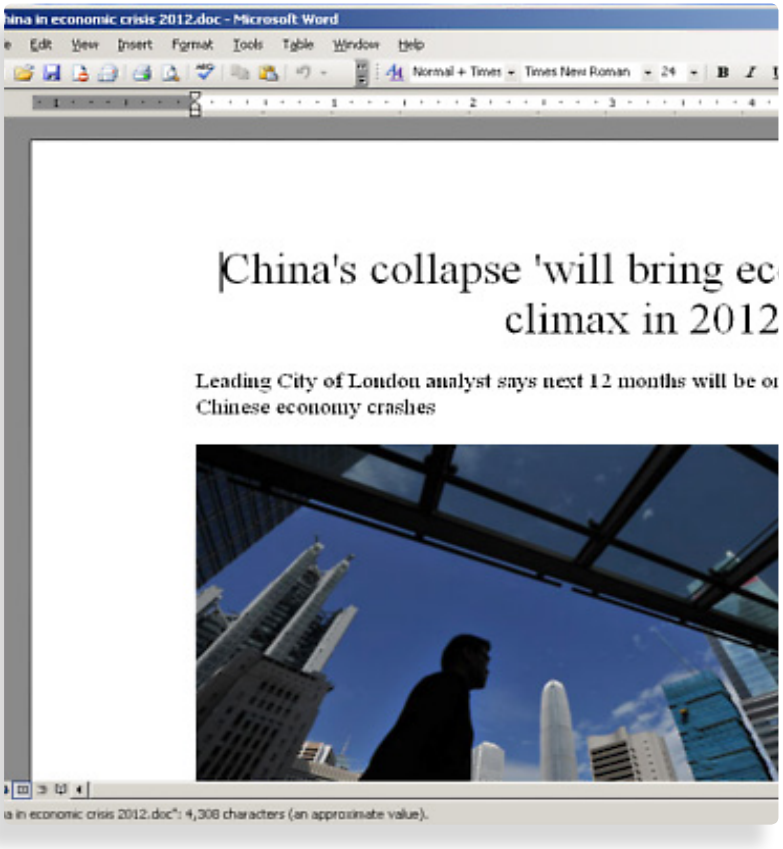


## TARGET DETAILS

From the very earliest variants we were able to attribute to Quedagh, we have noticed that their targets have been political in nature. Apart from other indicators, we can deduce the nature of the target based on the content of social engineering tactics used to distribute the installers. For example, one decoy dropped from a sample dating to 2012 (**image 2**) seems to be targeting European audiences and discusses a political/economic situation. Strings found in another sample from 2012 (**image 3**) again indicate a political motivation behind the attack. Most decoys used content taken from news sites; we noted one decoy dropped by an exploit document was created using the Russian version of Office (**image 4**).

The latest variant of the dropper pretends to be a document file with a Ukrainian filename (**image 5**). The choice of language for the filename again may tie in or reference the current political crisis in that country. The filename itself means 'password list' in English.

**Image 2:**
**Decoy document circa 2012**




## 2008 CYBERATTACKS ON GEORGIA?

During our investigation, we found interesting details that lead us to suspect that Quedagh might have been involved in the cyberattacks launched against Georgia during the 2008 Russo-Georgian confrontation. While the details identified are suggestive, they are not conclusive; they do however seem consistent with the group's involvement in subsequent targeted attacks.

## UKRAINE-RELATED PROXIES

While examining the samples collected during our BlackEnergy monitoring, we noticed that samples from this year had been updated to support the use of proxy servers while connecting to their C&C servers. This contrasts with earlier BlackEnergy 2 variants, which do not support proxy servers. In some network setups, a proxy server is needed to allow internal users to access the Internet [1]. BlackEnergy's use of a proxy server may indicate that the gang has prior knowledge of the target organization's internal setup to note of this requirement.

For example, in one sample the configuration uses the proxy server associated with the Ukrainian Railway (**image 6**). The configuration from another sample also shows it using an internal proxy under the giknpc domain (**image 7**). The domain giknpc.com.ua in turn hosts 3 domains (**image 8**), one of which is for the city of Dnipropetrovsk (**image 9**), the fourth-largest city in Ukraine, located in the southeast. Based on the set proxy servers for the different samples, we concluded that the gang is targeting Ukrainian government organizations.

**Image 3:**
**Strings from a sample circa 2012**

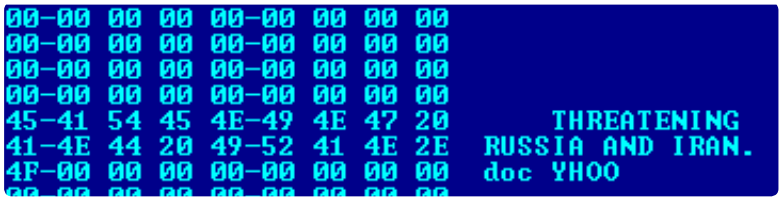


Image 4:
Decoy document created using a Russian version of Office

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<a:theme xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
name="Тема Office"><a:themeElements><a:clrScheme name="Стандартная"><a:dk1
```

Image 5:
2014 dropper sample disguised as a document. The filename means password list

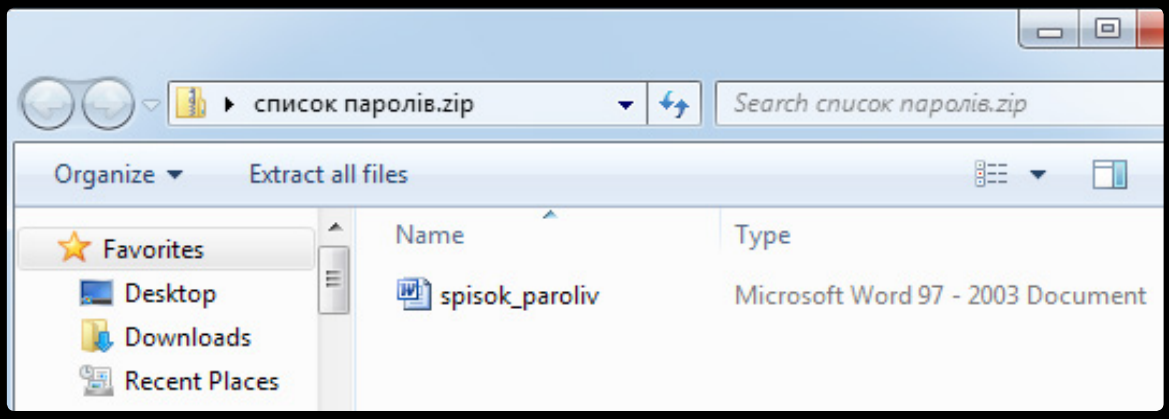



Image 6:
Configuration using Ukrainian Railway's proxy

```
82/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php;proxy=
tps=10.5.10.11:3128,https=proxy.pz.uz.gov.ua:3128,htt
=10.5.10.11:3129</addr>
```

Image 7:
Configuration using internal proxy under giknpc domain

```
<type>https</type>
<addr>https://5.79.80.166/templates/bf0dac805298cc1f633f19ce8ed6382f/
;proxy=https=172.20.124.10:8080,https=aproxy.giknpc.intranet:8080,htt
iknpc.intranet:8080,https=sproxy.giknpc.intranet:8080,https=nproxy.gi
net:8080,https=172.20.124.252:8080</addr>
</server>
```

Image 8:
Domains hosted on giknpc.com

## Network information

| | |
|---|---|
| IP address | 195.64.190.1 |
| Reverse DNS (PTR record) | relay.giknpc.com.ua |
| DNS server (NS record) | ns.giknpc.com.ua (195.64.190.1)<br>ns2.giknpc.com.ua (195.24.137.86) |

## Hosting information

| | |
|---|---|
| Number of domains hosted | 3 |
| Number of mail servers hosted | 1 |
| Number of name servers hosted | 1 |

Image 9:
Dnipropetrovsk domain

## Domains on 195.64.190.1

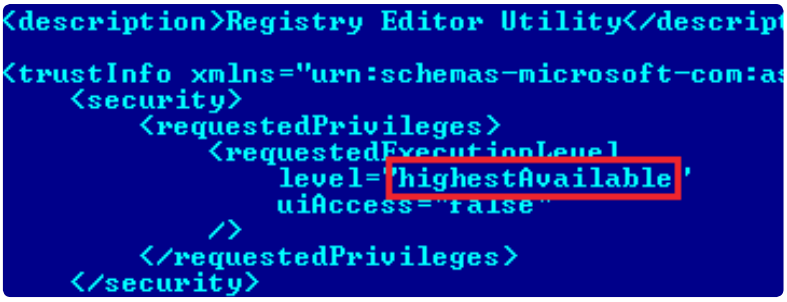
| | |
|---|---|
| adm.dp.ua | domain info |
| giknpc.com.ua | domain info |
| dp.gov.ua | domain info |

## TIMELINE

Although they may have started much earlier, the earliest BlackEnergy sample we could attribute to the Quedagh gang is from December 14, 2010.

Initially, the group seemed to prefer to use the filename of the Windows registry editor (regedt32.exe), presumably because the installer needs administrator rights to install its driver component and therefore would try to request for the highest available rights (**image 10**), if possible. As this triggers a notification message visible to the user, said user is more likely to grant permission if it appears to be the registry editor that is requesting for permission, since it is normal to run it with administrator rights. Experienced users though are less likely to be taken in, thereby decreasing the likelihood of a successful infection.

Image 10:
Installer requesting highest available rights




Starting April 2013, modified installers appeared showing that the Quedagh group found a way to bypass the default User Account Control (UAC) settings. With this change, the user's permission is no longer need (**image 11**). At this point, the gang also began to use the Windows installer program filename **msiexec.exe**.

## 64-bit support

Within a month of Windows 8.1's release, the gang had quickly added support for 64-bit systems, possibly anticipating that more of their target systems will be migrated to 64-bit systems. They also employ a neat trick to bypass the driver signing requirement on 64-bit Windows systems. As a side note, this latest finding updates and supercedes previously published research related to BlackEnergy's driver signing behavior [2].

However, this trick doesn't work on Windows 8 and later systems. The driver also crashes occassionally. This could be the reason for the stand-alone non-persistent BlackEnergy variant.

Image 11:
Installer execution privilege level amended


## BlackEnergy 3

We identified the new BlackEnergy 3 variant first by the change in its configuration, which differed from those of its two predecessors, 1 and 2 (**images 12 to 14**). It also no longer uses a driver component [3]. Further technical details are documented on page 10 to 11.

## TIMELINE OF BLACKENERGY & QUEDAGH HISTORY

**BLACKENERGY Development**

**BlackEnergy 1**

**BlackEnergy 2**

*May 12*
**BlackEnergy 3**

**Quedagh APT campaign**

*Dec 14*
First installer (regedt32.exe)

*Apr 9*
New UAC-bypassing installer (msiexec.exe)

*Nov 14*
64-bit support for BlackEnergy 2 driver

*Some time after Dec 25*
Targets Ukrainian entities

*CYBERATTACKS AGAINST GEORGIA*

*POLITICAL CRISIS IN UKAINE*

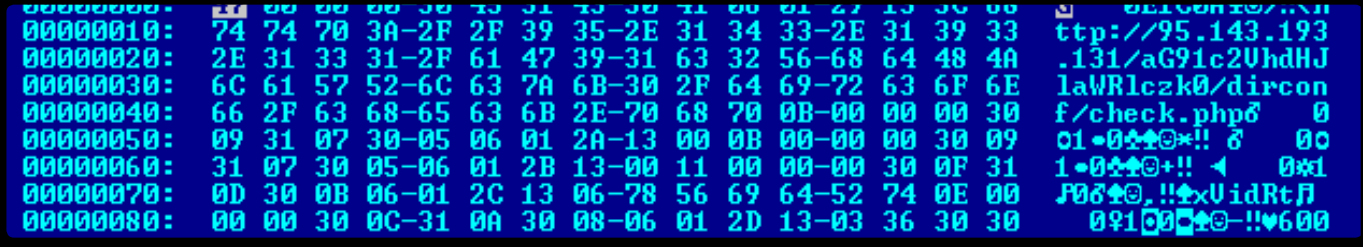
2007 2008 2009 2010 2011 2012 2013 2014

Image 12: BlackEnergy 1 configuration


Image 13: BlackEnergy 2 (aka BotnetKernel or bkernel [4] ) configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<bkernel>
<servers>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php</addr>
</server>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php;proxy=
https=10.10.255.55:3128,https=10.5.10.11:3128,https=proxy.pz.uz.gov.ua:3128,htt
ps=proxy.uz.gov.ua:3128,https=10.5.104.225:9080,https=10.5.10.11:3129,https=192
.168.1.22:3128</addr>
</server>
<server>
<type>https</type>
<addr>https://5.61.38.31/ZXBzaWxvbmVyaWRhbmkw/setattr.php</addr>
</server>
</servers>
<cmds>
</cmds>
<sleepfreq>600</sleepfreq>
<build_id>0D0B15aaa</build_id>
</bkernel>
```

Image 14: BlackEnergy 3 configuration



# TECHNICAL DETAILS

## UAC BYPASS DURING INSTALLATION

The malware will only attempt to infect a system if the current user is a member of the local administrator group. If not, it will re-launch itself as Administrator on Vista. This in effect will trigger a UAC prompt. On Windows 7 and later however, the malware will attempt to bypass the default UAC settings. It exploits a backward-compatibility feature found in newer versions of Windows. BlackEnergy installers include a Shim Database, or a "fix", instructing SndVol.exe to execute cmd.exe **(image 15, below)** instead in order to resolve the incompatibility.

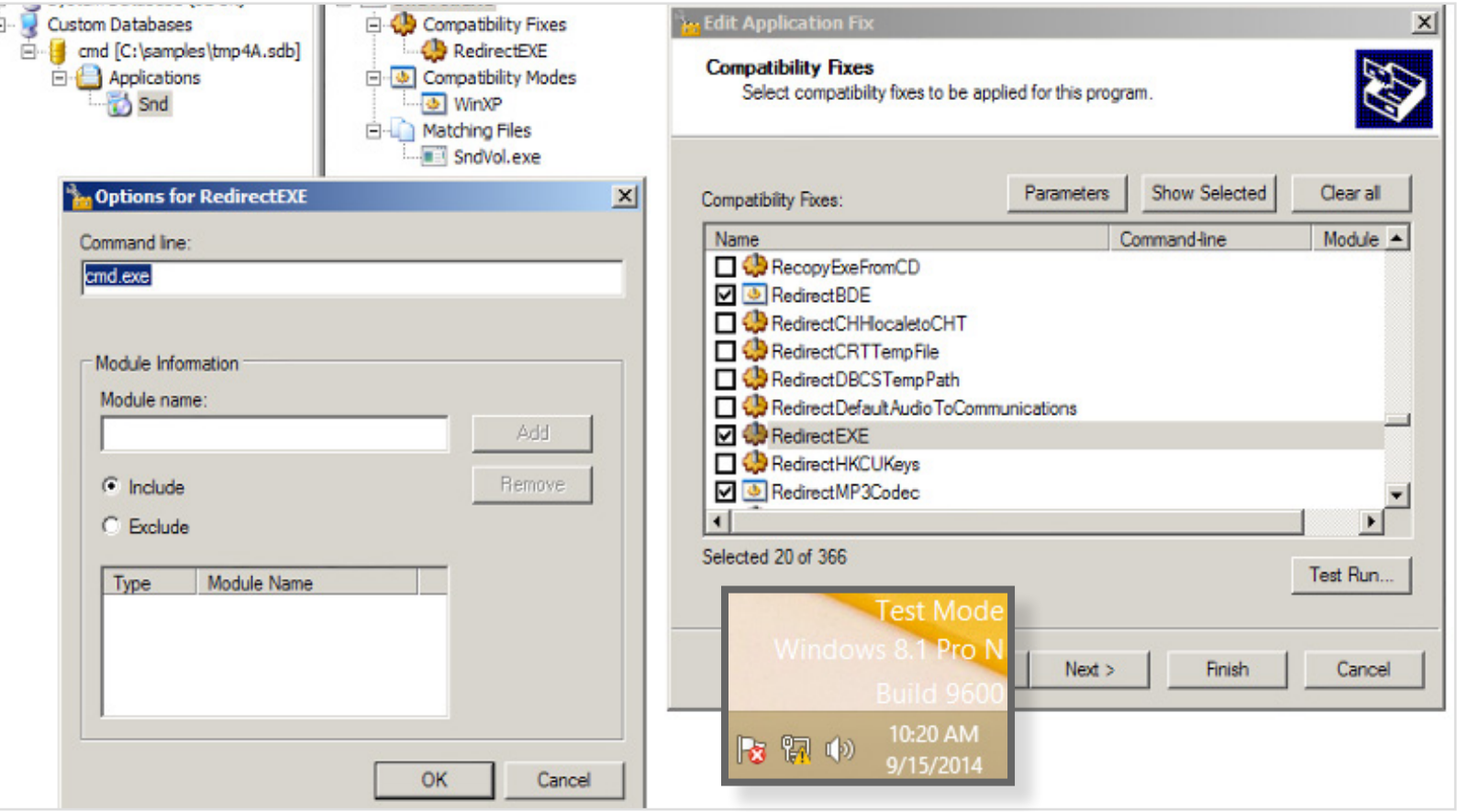
SndVol.exe is one of the Windows executables that will be automatically elevated upon execution because it is thought to be safe. What harm can a volume control cause? With the malicious "fix" installed however, executing SndVol.exe will execute the not-so-safe file cmd.exe instead, which can then be used to install the malware while in an elevated state.

## DRIVER SIGNING POLICY BYPASS

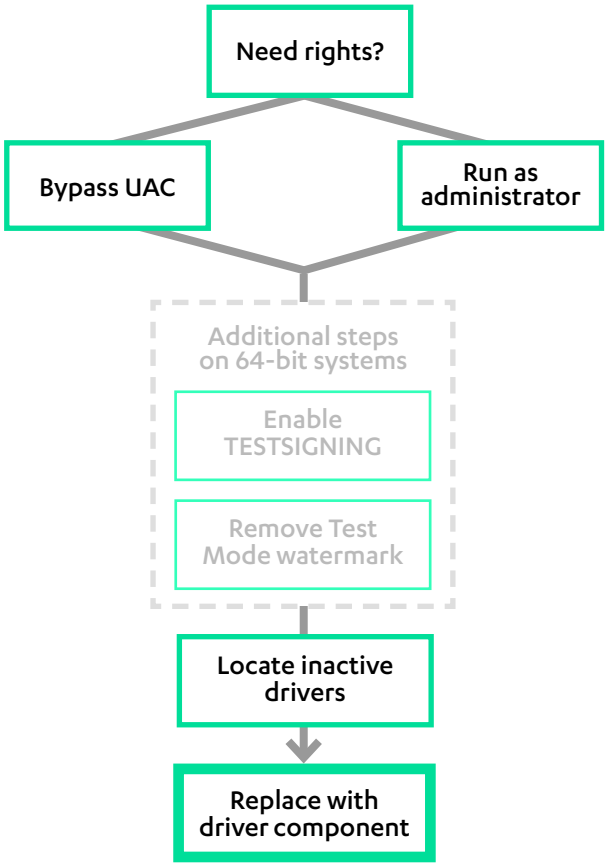
The role of the installer is to set up the malware's persistent component, which is the driver component. On 64-bit Windows systems, Microsoft has enforced a policy that requires all drivers to be signed as a security precaution. Signing provides a way to identify a driver to its author, effectively reducing the number of malware developers willing to take the risk. To allow developers to test their drivers during development, Microsoft provides a TESTSIGNING boot configuration option; while in this mode, a watermark is displayed on the screen to make it obvious to users and to prevent malware from exploiting this option.

BlackEnergy enables the TESTSIGNING option to load its driver component; to hide this change from the user, the malware removes the watermark by removing the relevant strings in the user32.dll.mui of the system. In Windows 8 and up however, the strings are no longer stored in user32.dll.mui, so the trick will not work. This may be one of the reasons for the existence of a standalone non-persistent BlackEnergy variant. The malware does not infect 64-bit Windows systems that are older than Vista.

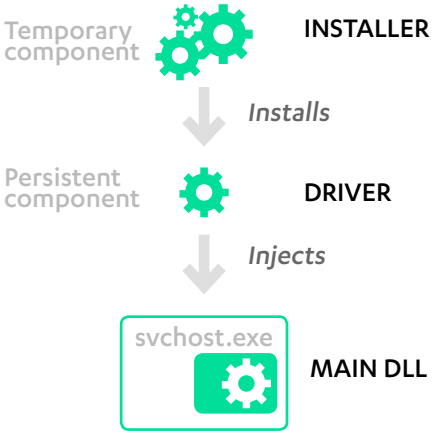
**Image 15: Malicious fix to redirect SndVol.exe to cmd.exe. Inset: Test Mode watermark**



### DIAGRAM 1: INSTALLATION FLOW




### DIAGRAM 2: ROLE OF DRIVER COMPONENT




### TABLE 1: IOCTL BUFFER COMMAND CODES

| Code | Function |
|---|---|
| 6 | Loads a driver into memory |
| 9 | Does not do anything; just returns true. Previously contained an uninstall routine. |
| 10 | Returns the registry path and driver file path |

## HIJACKING EXISTING DRIVERS

The installer will try to locate an existing driver service that is inactive. The service found will usually be a legitimate one that is disabled because it is no longer used or because it is set to start only on demand. The installer will drop the driver component using the corresponding path of the service. It will overwrite the existing driver if necessary. The hijacked service is then set to start automatically. This is how the malware is able to survive after a reboot. By doing this, the gang may be hoping that their malicious driver will be overlooked by administrators or investigators who are so used to seeing those legitimate services.

## DRIVER COMPONENT

The only component that will remain permanently on the infected system will be the driver component. The driver component used by the gang is a stripped down version of the BlackEnergy 2 driver.

The sole purpose of this driver component is to inject the main DLL component into svchost.exe. Interestingly, it does not contain the rootkit functionalities for hiding processes, files and registry objects that is found in the usual BlackEnergy 2 drivers. The gang may have opted for a 'hide in plain sight' approach to evade detections from rootkit scanners, such as GMER and RootkitRevealer, that checks for system anomalies.

The driver component provides a IOCTL interface to communicate with the main DLL component. **Table 1** (above) summarizes the command codes that can be passed to the IOCTL buffer. The 32-bit version contains additional, incomplete routines for hiding processes via direct kernel object manipulation (DKOM) and managing BlackEnergy 2 rootkit rules in memory [2].

**TABLE 2: TYPICAL BLACKENERGY DRIVER COMPONENT VERSUS QUEDAGH'S CUSTOM COMPONENT**

| | Typical BlackEnergy 2 | Quedagh BlackEnergy 2 |
|---|---|---|
| **Launch Point** | Creates a new service based on either a hardcoded or randomly generated name (depending on the installer) | Hijacks an existing legitimate service |
| **Role** | Hides processes, files and registry objects; Inject main DLL to svchost.exe | Injects main DLL to svchost.exe |
| **Versions** | 32-bit driver component that contains complete routines in its IOCTL interface | 32-bit driver component with a lot of remnant routines in its IOCTL interface, only a few of which make sense.<br><br>After Nov 11, 2013, the 64-bit driver component is available and provides limited functionalities in IOCTL interface (only those equivalent working routines found in the 32-bit versions) |

**TABLE 3: COMMANDS SUPPORTED BY VARIANTS TARGETED AT UKRAINIAN ENTITIES**

| Command | Description |
|---|---|
| **rexec** | Download and execute a binary |
| **lexec** | Execute a shell command |
| **die** | Uninstall |
| **getpl** | Load a plugin |
| **turnoff** | Quit (will start again after reboot) |
| **chprt** | Add / remove / set active command and control server |

BlackEnergy 2 was very well documented by Dell SecureWorks [5] in 2010. **Table 2** (above) summarizes the differences between the driver component used by Quedagh compared to the typical BlackEnergy 2.

## MAIN DLL COMPONENT

The core functionality of BlackEnergy 2 is found in the main DLL component. This component is embedded inside the driver component and is not found in the file system; this is to reduce the infection footprint on the system.

The main DLL provides a robust framework for attackers to maintain a botnet that is not tied to any specific functionality. The malware is designed to be used by loading customized plugins depending on the purpose of the botmaster. It is mainly a framework for plugins to communicate with a central command and control. Otherwise, the main DLL only provides a minimal set of commands. **Table 3** (above) summarizes the commands supported by the variants used in the attack against Ukrainian government organizations.

In BlackEnergy 2, the main DLL component communicates with its plugins via a defined set of API calls. It exports a number of function calls, which can be used by the plugins. On the other hand, plugins are required to export 2 functions to work. We highly recommend the research of Dell SecureWorks for those looking for more details regarding the BlackEnergy 2 plugin framework.

## BLACKENERGY 3

In contrast to previous variants, BlackEnergy 3 uses a simpler installer component. It does not have a driver component and the installer drops the main DLL component directly to the local application data (non-roaming) folder. The installer then creates a LNK file in the startup folder, using a filename generated based on the volume serial number as a launch point. The LNK file is a shortcut to execute the main DLL using rundll32.exe.

**TABLE 4: X509_ASN FIELDS & EQUIVALENT BLACKENERGY 2 XML NODE**

| ID | BlackEnergy 2 Node | Description |
|---|---|---|
| 1 | **servers** | The command and control servers |
| 2 | **plugins** | Plugins to be loaded |
| 3 | **cmds** | Commands to be executed |
| 4 | **build id** | Build ID |
| 5 | **sleepfreq** | Phone home interval |

**TABLE 5: BLACKENERGY 3 COMMANDS**

| Command | Description |
|---|---|
| **delete** | Uninstall |
| **ldplg** | Load a plugin |
| **unlplg** | Unload a plugin |
| **update** | Update main DLL |
| **dexec** | Download and execute an executable |
| **exec** | Download and execute a binary |
| **updcfg** | Update the configuration data |

This variant uses a new configuration format. The configuration data is a series of X509_ASN encoded values and are accessed by an ID number. **Table 4** summarizes the fields and their equivalent BlackEnergy 2 XML node, while **table 5** lists the completely new set of commands used in this latest variant.

BlackEnergy 3 also uses a different method of communication with its plugins, as it now communicates via RPC over the named-pipe protocol (ncacn_np).

## INFORMATION-STEALING PLUGIN

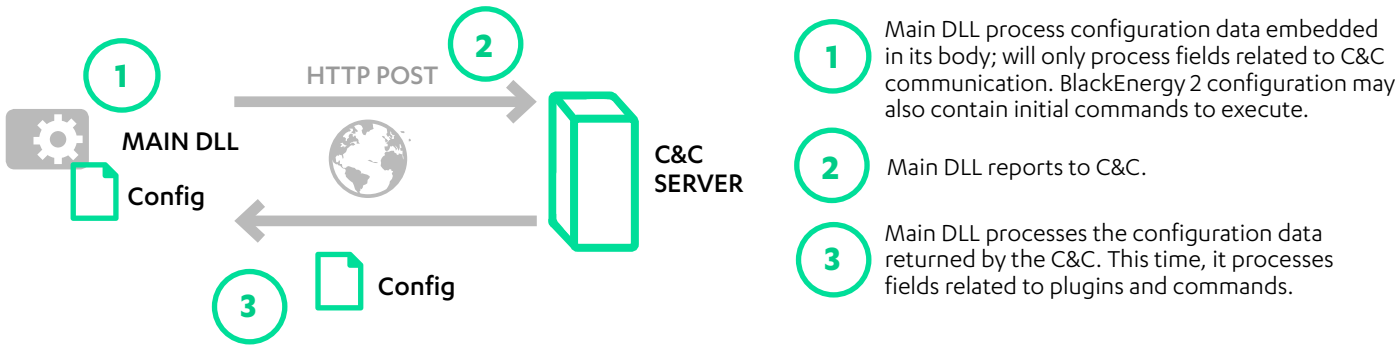
Since the main DLL component offers little clue as to what the malware was used for, we need to look at the plugin to determine the objective of the gang.

One particular plugin that was used in the campaign was called "si", perhaps to mean 'steal information'. The latest sample we found will attempt to gather the following information and send them to the C&C server:

- System configuration information (gathered via systeminfo.exe)
- Operating system version
- Privileges
- Current time
- Up time
- Idle Time
- Proxy
- Installed apps (gathered from uninstall program registry)
- Process list (gathered via tasklist.exe)
- IP configurations (gathered via ipconfig.exe)
- Network connections (gathered via netstat.exe)
- Routing tables (gathered via route.exe)
- Traceroute and Ping information to Google (gathered via tracert.exe and ping.exe)
- Registered mail, browser, and instant messaging clients (gathered via client registry)
  - Account and password information from The Bat! email client (gathered from account.cfn and account.cfg)
  - Stored username and passwords in Mozilla password manager of the following applications (gathered from signons*.txt and signons.sqlite)
    - Thunderbird
    - Firefox
    - SeaMonkey
    - IceDragon
  - Stored username and passwords in Google Chrome password manager of the following applications (gathered from "Login Data")
    - Google Chrome
    - Chromium
    - Comodo Dragon
    - Xpom
    - Nichrome
    - QIP Surf
    - Torch
    - YandexBrowser
    - Opera
    - Sleipnir
  - Account and password information from Outlook and Outlook Express
  - Internet Explorer version and stored username and passwords
  - Stored username and passwords in Windows Credential Store
    - Live
    - Remote Desktop
    - Other generic credentials (Microsoft_WinInet_*)

The nature of the information being gathered seems to be generic rather than targeted. This may be because the malware has roots from crimeware. The information is still useful however as such data makes it easier for the gang to plan any further attacks on the same targets.

### DIAGRAM 3: CONFIGURATION DATA HANDLING




### TABLE 6: MAIN DLL'S ADDITIONAL COMMANDS DURING DOWNLOAD OF ADDITIONAL FILES

| HTTP POST Field | Description of Values |
|---|---|
| **getp** | The plugin name to be downloaded |
| **plv** | Some variants specify the version of the plugin to be downloaded |
| **getpd** | The binary name to be downloaded |

## NETWORK TRAFFIC

BlackEnergy communicates with its C&C server via HTTP POST requests. For the BlackEnergy 2 samples used by the gang, the request contains the following fields:

**id=[bot_id]&bid=[base64_encoded_build_id]&dv=[x]&mv=[y]&dpv=[z]**

Where:

- bot_id is equivalent to the infected host name and the volume serial number following the format x[host_name]_[serial_no] (e.g. xJOE-PC_484DA98A)
- build_id is the string found in the build_id field in the sample's configuration data
- x, y, z are hardcoded values which varies among samples

The fields are almost the same for BlackEnergy 3 samples:

**id=[bot_id_sha1]&bid=[base64_encoded_build_id]&nm=[x]&cn=[y]&num=[z]**

The only major difference is that the id field contain just the hash instead of the actual string. The actual bot_id string in which the id hash is derived is also a bit different; it now uses the format [domain_sid]_[host_name]_[serial_no].

The response of the command and control server will be encrypted using the id field in the POST request as the key. After the response is decrypted, it will be in the form of the corresponding configuration data of the BlackEnergy sample; for example, BlackEnergy 2 samples expect the decrypted response to be a XML document, while BlackEnergy 3 samples expect the decrypted response to be a series X509_ASN encoded values.

The decrypted response, which is equivalent to another configuration data, will be processed similar to the initial configuration data embedded in the main DLL; the only differences are the data fields that are processed. This cycle is illustrated in **diagram 3** (above).

The main DLL also uses the fields listed in **table 6** (above) when it needs to download additional files.

## CONCLUSIONS

**BlackEnergy** is a toolkit that has been used for years by various criminal outfits. In the summer of 2014, we noted that certain samples of BlackEnergy malware began targeting Ukranian government organizations for information harvesting. These samples were identified as being the work of one group, referred to in this document as "**Quedagh**", which has a history of targeting political organizations. Though inconclusive, suggestive details indicate that BlackEnergy malware, possibly also from this gang, may also have been used in the Russo-Georgian confrontation in 2008.

The Quedagh-customizations to the BlackEnergy malware include support for proxy servers (which, in the samples examined are associated with Ukrainian entities) and use of techniques to bypass User Account Control and driver signing features in 64-bit Windows systems. While monitoring BlackEnergy samples, we also encountered a new variant, which we dub **BlackEnergy 3**, with a modified configuration, no driver component and a different installation procedure.

The use of BlackEnergy for a politically-oriented attack is an intriguing convergence of criminal activity and espionage. As the kit is being used by multiple groups, it provides a greater measure of plausible deniability than is afforded by a custom-made piece of code.

## REFERENCES

1. Wikipedia; Proxy server; http://en.wikipedia.org/wiki/Proxy_server#Cross-domain_resources
2. Broderick Aquilino; F-Secure Weblog; *BlackEnergy Rootkit, Sort Of*; 13 June 2014; http://www.f-secure.com/weblog/archives/00002715.html
3. Broderick Aquilino; F-Secure Weblog; *Beware BlackEnergy If Involved In Europe/Ukraine Diplomacy*; 30 June 2014; http://www.f-secure.com/weblog/archives/00002721.html
4. Kafeine; Malware don't need Coffee; *BotnetKernel (MS:Win32/Phdet.S) an evolution of BlackEnergy* ; 21 June 2014; http://malware.dontneedcoffee.com/2014/06/botnetkernel.html
5. Joe Stewart; DELL SecureWorks; *BlackEnergy Version 2 Analysis*; 3 March 2010; http://www.secureworks.com/cyber-threat-intelligence/threats/blackenergy2/

# APPENDIX A | SAMPLES

| SHA1 | Description |
|---|---|
| 26b9816b3f9e2f350cc92ef4c30a097c6fec7798 | Main reference for related BlackEnergy 2 32-bit driver and main DLL component analysis |
| bf9937489cb268f974d3527e877575b4fbb07cb0 | Main reference for related BlackEnergy 2 64-bit driver (signed on 2013-12-25) and installer analysis. Basis for the start of the Ukrainian target. |
| 78636f7bbd52ea80d79b4e2a7882403092bbb02d | Main reference for related BlackEnergy 3 analysis |
| bf9172e87e9264d1cddfc36cbaa74402bb405708 | Main reference for related si plugin analysis |
| 441cfbaba1dfd58ce03792ef74d183529e8e0104 | Stand-alone non-persistent BlackEnergy 2 sample |
| f7d4aa90b76646f4a011585eb43b9d13c60f48eb | Trojanized Juniper installer containing related BlackEnergy 2 |
| 8ccd2962bce8985d0794daed6e0bf73e5557cfe8 | Trojanized Adobe Bootstrapper containing related BlackEnergy 2. This means that it is highly probable that there is a trojanized Adobe package out there. |
| d496f99f7e07d5cbbd177a9d43febe8fb87ebc3b | Related RTF document containing exploit |
| cc71aa8f919911676fb5d775c81afc682e6e3dd3 | Related BlackEnergy 2 binary containing strings that are political in nature |
| abab02d663872bcdbe2e008441fcd7157c0eb52d | Oldest (compiled on 2010-12-14) related BlackEnergy 2 installer that was found |
| e5c8c10b10ee288512d3a7c79ae1249b57857d23 | Oldest (compiled on 2013-04-09) related BlackEnergy 2 installer that bypass UAC that was found |
| 8743c8994cc1e8219697394b5cb494efa7dad796 | Oldest (signed on 2013-11-14) related BlackEnergy 2 64-bit driver that was found |
| 285b3252a878d1c633ea988153bbc23c148dd630 | Oldest (compiled on 2014-05-12) related BlackEnergy 3 dropper that was found |

**PAGE INTENTIONALLY LEFT BLANK**

# SWITCH ON FREEDOM

F-Secure is an online security and privacy company from Finland. We offer millions of people around the globe the power to surf invisibly and store and share stuff, safe from online threats.

We are here to fight for digital freedom.

Join the movement and switch on freedom.

Founded in 1988, F-Secure is listed on NASDAQ OMX Helsinki Ltd.

F-Secure.