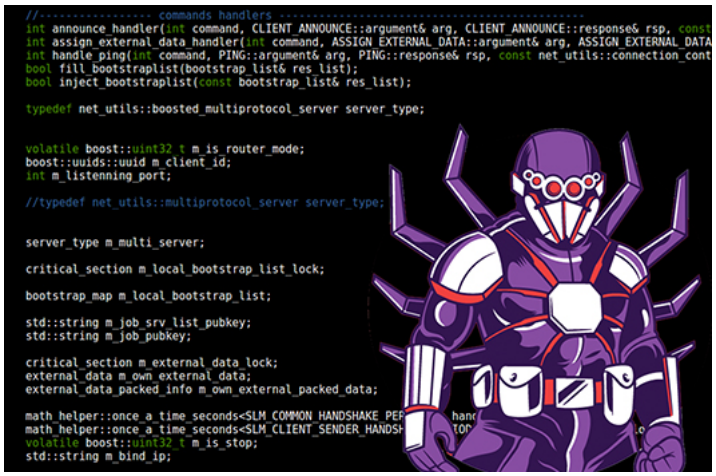
Farewell to Kelihos and ZOMBIE SPIDER

crowdstrike.com/blog/farewell-to-kelihos-and-zombie-spider/

Brett Stone-Gross, Tillmann Werner, and Bex Hartley

December 5, 2018



The Kelihos peer-to-peer botnet was one of the largest and longest-operating cybercrime infrastructures in existence. Its origins can be traced back to the Storm Worm, a botnet that emerged in 2007 and was one of the earliest criminal malware infrastructures to leverage peer-to-peer technology. After the demise of Storm, it was replaced by another new botnet known as Waledac that also leveraged peer-to-peer communications. Waledac was taken over and neutralized by a group of researchers in September 2010[1]. The first generation of Kelihos emerged in December that year, three months after its predecessor Waledac was dismantled.

Kelihos itself was subject to several[2] takeover[3] operations[4], each of which lead to the botnet being rebuilt in a new, more robust manner. The fifth and last generation of the botnet had been around since summer 2013, with an estimated size of 40,000 infected machines. It was neutralized by the U.S. Department of Justice with technical assistance by CrowdStrike in April 2017[5].

The Kelihos malware featured a wide assortment of plugins for different criminal purposes but was primarily used to deliver spam emails. Its peer-to-peer network protocol was designed to be difficult to reverse engineer, containing several layers of encryption, including RSA, Blowfish and a custom obfuscation algorithm that the malware author referred to as “monkey” functions in the code. This design is a clear reaction to previous takedowns with the goal to raise the bar for future attacks, but it ultimately failed to protect the botnet against attacks.

The primary threat actor, who was tracked by CrowdStrike as ZOMBIE SPIDER, rose to prominence in the criminal underground under the moniker *Peter Severa*. The individual behind this handle is *Peter Yuryevich LEVASHOV*[6] who was arrested in Spain when the final version of Kelihos was taken over in April 2017, and who recently pleaded guilty to operating the botnet for criminal purposes[7].

The purpose of this blog is to summarize and share our findings about Kelihos and its operator. The first section summarizes the results of our technical analysis of the Kelihos malware. The second section discusses attribution and provides some context around the threat actor. The blog concludes with an outlook section and we provide a YARA rule for detection in the Appendix.

Technical Analysis of Kelihos

Modern spam botnets have to be flexible in the way they run campaigns in order to be able to quickly adapt to new detection techniques. Kelihos, like many others, implemented a sophisticated spam engine that automatically constructs spam messages from templates and additional inputs to avoid any patterns that can be used in filters. Despite the flexibility provided by the template system, some spam campaigns exhibited recurring characteristics and several researchers believed that there existed multiple simultaneously operated versions of the botnet. This was never the case.

Spam jobs that were distributed by the botnet operator defined a message template. A bot would populate this template with randomly generated strings or information taken from additional dictionary files that contained, for example, subject lines or URLs. A captured spam template is shown below, with several variable fields highlighted in different colors.

```
 Received: from %^C0%^P%^R3-6^%:qwertyuiopasdfghjklzxcvbnm^%^% ([%^C6%^I^%.%^I^%.%^I^%.%^I^%^%])
      by %^A^% %^Fsendmailver^% with SMTP id %^Y%^C5%^R20-300^%^%^%037036;
      %^D%^V5^%^%
Message-ID: <%^O%^V6^%:%^R3-50^% %%^V0^%>From: "%^C4%^Fmynames^%^%" <%^Fnames^%@%^Fdomains^%>
To: <%^0^%>
Subject: %^Fpharma^%
Date: %^D-%^R30-600^%^%
MIME-Version: 1.0
Content-Type: text/plain;
      format=flowed;
```

```
      charset="%^Fcharset^%";
      reply-type=original
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.%^C7%^Foutver.6^%^%
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.%^V7^%
```
%^J%^Fpharma^% %^Fmirabella_links2^%^%

The following is an email constructed from this template.

```
Received: from iaw ([232.59.54.125])
       by ppp-188-174-39-206.dynamic.mnet-online.de (8.13.1/8.13.1) with SMTP id 201104051045037036;
       Tue, 5 Apr 2011 10:45:55 +0100
Message-ID: <002101cbf36d$426b6370$e83b367d@seclabiaw>
From: "Christina" <bcchiang@parteck.net>
To: <[redacted]>
Subject: Wonderful revealing effect on your libido.
Date: Tue, 5 Apr 2011 10:32:16 +0100
MIME-Version: 1.0
Content-Type: text/plain;
       format=flowed;
       charset="iso-8859-1";
       reply-type=original
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2900.2180
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2180

Bring more enjoyment to your life, get a magicpil!
http://drokkies[.]nl/dwg2c4v.html
```

For several years, pump-and-dump stock scams, dating ruses, credential phishing, money mule recruitment and rogue online pharmacy advertisements were the most common spam themes. In 2017, however, Kelihos was frequently used to spread other malware such as *LuminosityLink*, *Zyklon HTTP*, *Neutrino*, *Nymaim*, *Gozi/ISFB*, *Panda Zeus*, *Kronos*, and *TrickBot*. It was also observed spreading ransomware families including *Shade*, *Cerber*, and *FileCrypt2*.

Malware Distribution

The Kelihos malware distribution model involved affiliates of a pay-per-install service operated by ZOMBIE SPIDER. Each affiliate was provided with a custom malware binary with a unique tag hard-coded into the executable. The criminal operators of Kelihos were able to track and credit affiliates for infections based on these tags when the malware communicated with their backend infrastructure.

Compared to other malware families, Kelihos executables are relatively large due to the use of several third-party libraries, including *Crypto++* for handling encryption-related functions, the *Boost* library that provides a wide variety of convenience functions, and the *WinPcap* library that is used for capturing credentials used in plaintext network protocols.

Affiliates frequently distributed Kelihos executables through social engineering and exploit kits. In addition, the Kelihos peer-to-peer network provided a fast-flux DNS hosting service that was often used in combination with spam campaigns to serve its own binaries. As an example, the URL `http://betaler[.]com/gl1_1.php` was hosted by that fast-flux service network. In this case, the content served from this URL was some simple JavaScript-based redirect code shown below:

```
<!DOCTYPE HTML><html><head><script type="text/javascript">parent.location.href="http://combach[.]com/adobe/";
</script></head><body></body></html>
```

The domain `combach[.]com` from the redirect target was hosted on the Kelihos fast-flux service network as well. Visiting users were presented with the fake Adobe Flash Player website shown in Figure 1 in an attempt to deceive them into clicking the installation link, which would, in turn, provide a Kelihos malware executable.

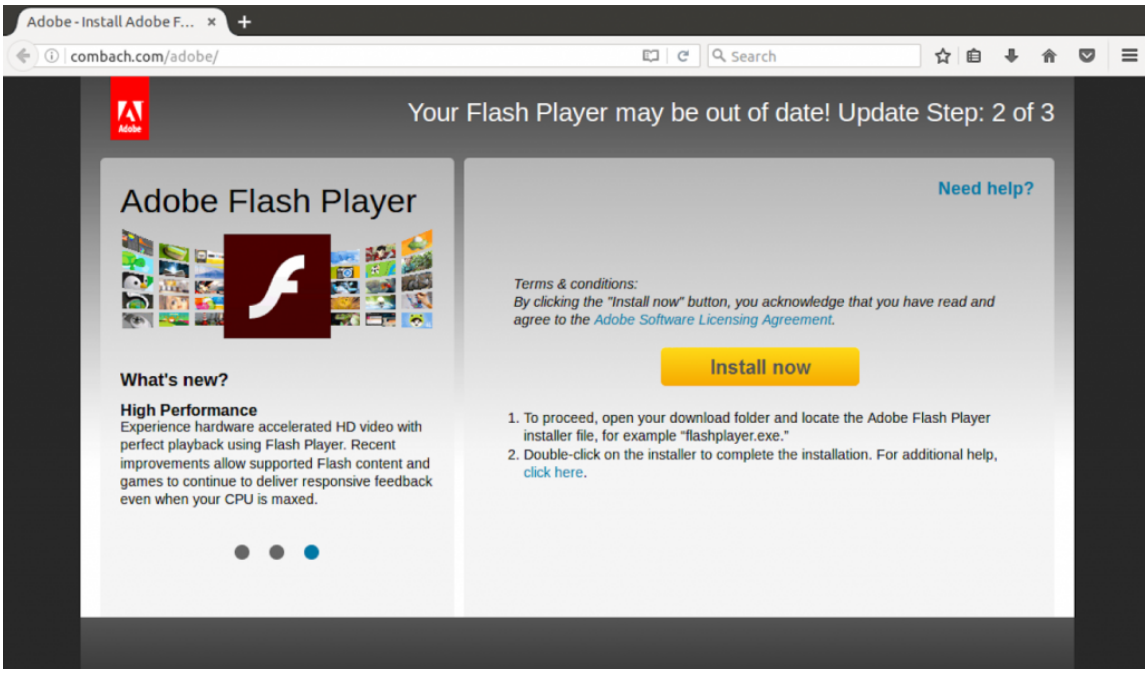


Figure 1. Fake Adobe Flash Player Installer Website

Installation and Persistence

The malware establishes persistence by creating a registry name and value pair under the key `Software\Microsoft\Windows\CurrentVersion\Run` in the `HKEY_LOCAL_MACHINE` hive if the user has administrator privileges, or the `HKEY_CURRENT_USER` hive, otherwise. The registry name consists of a word from the prefix noun list shown below concatenated with a word from an action suffix list. Its value points to the Kelihos executable on disk. Kelihos modifies the file attributes on its own executable to *hidden* and *read only*. The following prefix nouns were used:

- `Connection`
- `CrashReport`
- `Database`
- `Desktop`
- `Folder`
- `Icon`
- `Media`
- `Network`
- `Time`
- `Tray`
- `Video`

The following is the list of suffix nouns used to construct the name string:

- `Checker`
- `Informer`
- `Notifyer`
- `Saver`
- `Updater`
- `Verifyer`

Upon initial infection, the malware generates a 16-byte unique bot identifier that is used during peer-to-peer communications. This value is created from 15 randomly generated bytes plus a single-byte checksum that is computed by adding the 15 random bytes together.

All Kelihos binaries start with a list of hard-coded peers to bootstrap the process of joining the peer-to-peer network. All analyzed samples had dozens of such hard-coded entries, each consisting of an IP address, a TCP port number (which in all cases is 80), the last time a peer has been contacted (which defaults to 0 in the bootstrap list), a bot ID, and the number of seconds a peer has been live, also defaulting to 0.

The peer list is stored in the Windows registry with the name determined by concatenating strings from three dictionaries. However, due to a bug in the code, this name will always be `DBSavedUse` when the malware is executed for the first time. The value stored at this name always starts with the magic byte pattern `A2 49 4D F3 D9 1E 9F 88 01` that is used as a signature to identify serialized data and also present in each peer-to-peer protocol message. In addition to the peer list, Kelihos will create three more name/value pairs under this registry key that

store (1) a master key value, (2) the last job ID value, and (3) the bot ID value encoded with Base64. Due to the bug mentioned earlier, these registry names will always be `PersistentLocalizedName`, `PlatformCompressedValid`, and `LineLoadedQuick .` In addition, if Kelihos is running in router mode (see below), the registry name `RecordEnabledCheck` will also be created.

Despite the bug in the code, identifying the registry key that stores the Kelihos configuration information is non-trivial. The precise location of the registry key is selected by computing a histogram of the character length and the uppercase and lowercase frequencies for each key and subkey in the `HKEY_CURRENT_USERS` hive. The results of the histogram are then sorted, and the first entry in the list is chosen to hold the configuration information. Consequently, different infected machines will likely store the data in different locations.

Peer-to-Peer Protocol

Infected machines form a peer-to-peer network with a hierarchical architecture shown in Figure 2. There are three tiers, referred to as *job servers*, *router nodes*, and *worker nodes*. When a system is infected, the malware checks the network adapter settings to determine whether it has a publicly routable IP address. If that is the case, the bot will start in the router mode of operation and create network services on TCP port 80 for peer-to-peer communications and on UDP port 53 for participating in the fast-flux network. If the system has no public IP address, the malware will start in worker mode and receive tasks to generate spam emails.

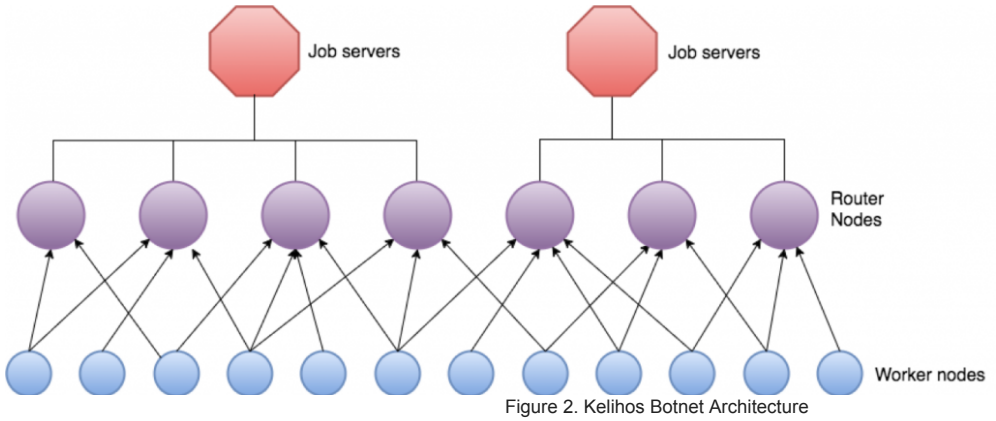


Figure 2. Kelihos Botnet Architecture

There are two primary types of peer-to-peer messages: peer lists and jobs. All peer-to-peer communications occur over TCP port 80, with peer lists being exchanged over a custom binary protocol and jobs being distributed using the same protocol with the addition of HTTP encapsulation. This distinction comes from the fact that messages related to tasking are being processed by the HTTP-based backend servers, whereas all other messages are exchanged between nodes that are part of the dynamic, self-organizing, peer-to-peer network — there is no need to encapsulate these in HTTP sessions.

The custom network protocol used for all message types makes use of RSA to perform a key exchange among peers and subsequently encrypt data with this session key. The first packet of the key exchange is similar to the following:

```
00000000  d5 e2 57 60 6c 55 55 45  03 10 48 40 99 5b 9f ad ..W`lUUE ..H@.[..
00000010  72 1e 36 2f 44 e1 00 0c  16 dd 9e 04 30 46 02 41 r.6/D... ....0F.A
00000020  00 d0 5f a9 4d e0 34 a9  21 c8 e4 30 43 47 aa 7a .._.M.4. !..0CG.z
00000030  00 6f ea 0d a4 8f d6 3f  b1 c9 6b c9 c4 93 54 5f .o.....? ..k...T_
00000040  d7 70 1a de 1c b1 5c 4d  ca cf 61 86 14 a4 31 63 .p....\M ..a...1c
00000050  75 60 9e 9b 69 b4 8e d7  19 26 1f 56 66 49 ab bd u`..i... .&.VfI..
00000060  e3 02 01 11 79 e2 f6 4d  f4 56 c1 22 6c 23 90 3a ....y..M .V."l#.:
00000070  60 4f be 69 a3 78 f2 a0  bc c5 ff ca 99 c7 7c 18 `O.i.x.. ......|.
00000080  1b 65 26 2b 0f dd 1b e6  3a f4 13 e0 64 bf 25 89 .e&+.... :...d.%.
00000090  86 ba e2 1f 5d d0 f1 06  e8 71 2e ea a5 b8 64 ef ....]... .q....d.
000000A0  ae bf 8d a7                                      ....
```

The first DWORD in the hexdump above specifies the protocol version. Its value has been generated by a bit-scrambling function with random entropy to obfuscate the actual version number, which was 5 in the last generation of the botnet. The second DWORD is the size of the message, which is obfuscated using another bit-scrambling function. The four bytes at offset 8 serve as a header for the payload data, which is composed of serialized blocks. This header has the following structure:

1. Number of blocks (0x03)
2. Size of the first block: 16-byte session key (0x10)
3. Size of the second block: peer's RSA public key in BER format (0x48)
4. Size of the third block: RSA signature of the 16-byte session key (0x40)

The remote peer responds with a message similar to the following:

```
00000000  1a 28 72 06 f2 55 55 45  02 40 a1 01 40 b7 fd 8e .(r. .@..@...
00000010  e0 d1 88 4f ab cd 1d c3  fc e5 bf e2 5f 03 46 3f ...O.... ...._.F?
00000020  2f f3 43 92 67 15 ac ed  3c 68 49 88 27 55 5a b5 /.C.g... <hI.'UZ.
```

```
00000030  cf a4 92 c2 38 74 27 12  a8 1e e7 62 ef 63 49 9b ....8t'. ...b.cI.
00000040  e9 4f 85 3c 69 1f d2 b6  d8 e6 52 38 04 88 3a 93 .O.<i... ..R8..:.
00000050  41 b0 f8 b6 ef e0 a7 64  68 47 70 1a 2c 86 b7 93 A......d hGp.,...
00000060  55 cd d3 c2 c5 8d b0 39  24 7c 20 bd 8d c4 35 10 U......9 $| ...5.
00000070  97 73 1d 1c 0a 3c 29 92  8c 30 b6 bf ac cf a2 61 .s...<). .0.....a
00000080  92 40 61 e7 06 32 11 74  41 c9 1c 3b b5 9f 2d c4 .@a..2.t A..;..-.
00000090  d4 64 4e 04 e6 8f d9 69  27 e2 0a ae 6c 12 d8 59 .dN....i '...l..Y
000000A0  3f 06 97 92 04 39 88 9b  57 1d cf 49 7f 78 ce 0e ?....9.. W..I.x..
000000B0  ef b3 ea 31 3d f9 44 c0  0a 30 ca e2 f4 50 84 0b ...1=.D. .0...P..
000000C0  2a d7 34 b8 cb 5d 11 70  52 4f 86 76 3e 6e b4 e1 *.4..].p RO.v>n..
000000D0  94 a5 b0 94 2e 7c 7e 9b  d6 41 ad 0b 48 3c 8b b0 .....|~. .A..H<..
000000E0  60 d9 a3 1b 19 c7 84 d7  1f ac 97 5e 1e          `....... ...^.
```

This packet contains the same header structure as described above, with the payload consisting of a Blowfish key encrypted with the local peer's RSA public key. This Blowfish key is used to decrypt the second data block, which contains a block structure that is identical to the initial request, containing the session key, the remote peer's RSA public key, and the remote peer's signature of the 16-byte session key. If all values match, the cryptographic session has been successfully established.

After the public key exchange is complete, Kelihos serializes subsequent messages and uses the session key to determine a sequence of "monkey" function calls in order to scramble and thus obfuscate the payload. Finally, the message is encrypted using a random 16-byte Blowfish key that is, in turn, encrypted with the remote peer's RSA public key.

Each peer-to-peer protocol message has a type identifier. The following list shows existing message types and their purpose:

- 0 job task message
- 1000 peer list exchange message
- 1002 job request message
- 1003 ping request
- 1004 pong response
- 1005 email harvest results message

The inner layer of the peer-to-peer protocol is a serialized message format processed by a library called *ANMP* that was later published as open-source software (see the Attribution section). This library implements a basic run-length encoding of primitive data types like integers, lists, lists of strings, maps and binary objects. Complex data types are supported in the form of vectors that combine multiple primitives. The serialization and deserialization code is relatively sound and of better quality than most malware code, however, it contains some mistakes that can result in crashes or worse.

Serialized data is reduced in size with the Lempel-Ziv compression algorithm before being encrypted with the Blowfish cipher in CBC mode. We developed our own code to be able to parse and generate Kelihos peer-to-peer protocol messages. The following sections include the output of several parsed messages as generated by our tools to visualize their structure.

Peer List Messages

The most important part of any peer-to-peer network is the ability for peers to exchange lists of other nodes in order to share information about nodes that have joined the network with neighbors. This allows the network to self-organize dynamically in order to maintain connectivity between peers. It further eliminates a single point of failure as there is no central control instance, which makes the network more resilient to disruption and takeover attempts.

Each Kelihos peer exchange can process up to 500 entries, although peers maintain lists of up to 3,000 entries. Bots initiate a peer list exchange by sending their current peer list to another node. The receiving node will merge the remote peer's list with its own and construct a response from the results. This design exhibits a fatal weakness that allows for active propagation of fake information in the peer-to-peer network.

Peer list exchange messages contain an `m_external_info_packed` field that is digitally signed with RSA. This field contains a list of IP addresses and ports for job servers — central systems that are controlled by the botnet operator. This information is critical for router nodes to know where to proxy traffic upstream. The protocol also supports a field for list of special router node entries, called `m_trusted_routers`, that is also digitally signed with RSA; however, we never observed populated values in this field. The list appeared to be designed as a resiliency measure to regain control of a router node by replacing its peer list with entries controlled by the Kelihos threat actor. Since whoever controls either of these fields can easily take over the network, they were protected against misuse by digital signatures. An example of a parsed peer list is shown below (some fields are truncated for readability):

```
 m_client_id (1): cf735914-32ed-4aef-bbea-13237b7525f7
m_current_time (3): 2017-03-31 19:58:07 GMT (58deb4cf)
m_success (4): 22626
m_bootstrap_list (5):
   m_clients_list (j): (500 elements)
        m_ip (b): 176.223.45[.]2
       m_listening_port (d): 80
       m_last_active_time (g): 2017-03-31 19:03:40 GMT (58dea80c)
```

```
      m_client_id (p): 3b7813a4-c61f-43d0-8351-cc6b0765bf98
      m_live_time (x): 468         m_ip (b): 100.82.77[.]2
      m_listening_port (d): 80
      m_last_active_time (g): 2017-03-31 19:05:01 GMT (58dea85d)
      m_client_id (p): 42639ffe-2a48-4a61-9ec7-6f6340e9db9a
      m_live_time (x): 2676"
   m_last_start_build (a): 0
m_current_time (y): 1970-01-01 00:00:00 GMT (00000000)
m_listening_port (s): 0
m_real_target_ip (v): 178.56.138[.]29
m_external_info_packed (vf):
   m_external_info_id (g): 2017-02-16 08:05:17 GMT (58a55d3d)
    m_external_data_blob (d): (encrypted blob)
      m_job_servers (2): (4 elements)
         m_ip (2): 194.165.16[.]66
         m_port (6): 80             m_ip (2): 194.165.16[.]69
         m_port (6): 80             m_ip (2): 91.195.103[.]13
         m_port (6): 80             m_ip (2): 91.195.103[.]14
         m_port (6): 80
      m_list_id (7): 2017-02-16 08:05:17 GMT (58a55d3d)
   m_external_data_signature (h): string (256 bytes):
      0x00000000  65 29 8e 69 a7 59 e5 37  f8 37 73 49 0c 44 2b 31
      0x00000010  db 2f bb 12 d4 96 85 bf  cc 76 0c 4a 08 7f d6 f9
      ...
m_trusted_routers (0):
   m_external_info_id (g): string (8 bytes): 71f010f05b6bdcfe
    m_external_data_blob (d): string (0 bytes):
   m_external_data_signature (h): string (0 bytes):
```

Golden Parachute Domain

Every Kelihos sample had a hard-coded domain name that the author referred to as a “golden parachute domain.” The purpose of this domain is to assist a bot in regaining access to the peer-to-peer network if no nodes in its peer list are reachable. This may occur naturally if an infected system has been offline for a period of time. The golden parachute domain is hosted by the Kelihos fast-flux DNS service and resolves an IP address of a current router node, which is sufficient for the bot to bootstrap and regain connectivity with the network. The last golden parachute domains we observed were `gorodkoff[.]com` and `goloduha[.]info` . Resolution attempts for these may be indicative of a Kelihos infection.

Job Messages

Kelihos provides configuration information and commands to infected systems through job messages. Requests for jobs are sent as HTTP requests using a word from the path list shown below, appended with an `.htm` extension.

- `default`
- `file`
- `home`
- `index`
- `install`
- `login`
- `main`
- `online`
- `search`
- `setup`
- `start`
- `welcome`

An example Kelihos job request appears similar to the following:

```
 GET /file.htm HTTP/1.1
Host: 103.229.85.197
Content-Length: 7515
User-Agent: Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1467.0
Safari/537.36 [removed binary data]
```

This HTTP request is anomalous, as it combines the GET method with binary data in the body. To make network signature generation more difficult, Kelihos randomizes the User-Agent string from a list that is hard-coded into the malware.

Download-and-Execute Commands

Kelihos has the ability to push arbitrary second-stage payloads to infected systems. This feature can be used to update the binary or deliver additional malware. The command provides a digitally signed URL to prevent unauthorized delivery of payloads.

Spam Jobs

As mentioned above, the primary purpose of Kelihos is to distribute spam emails. The botnet operates as a service that allows other criminals to pay to deliver their own spam. The behavior of the Kelihos SMTP spam engine is controlled by the `m_mail_section` configuration section in a job task. This provides the infected system with all the instructions necessary to carry out a spam campaign, such as the email address target list, name servers for MX records, and the email content. Kelihos nodes running in worker mode (those with a non-public IP address) conduct direct SMTP spam activities. This prevents Kelihos router nodes from getting blacklisted by anti-spam services. An example spam template configuration is show below (some fields have been removed or truncated for brevity):

```
m_mail_section (b3):
    m_generic_config (g):
        m_threads_count (c): 0
        m_send_queue_max_size (v): 20000
        m_reports_mode (b): 0
        m_adress_per_client (n): 10000
        m_smart_mailing (s): 1
        m_smart_threads_max (f9): 0
        m_smart_level (k): 70
        m_sockets_max (p): 1024
        m_errors_num_to_ban (e): 300
        m_sleep_when_ban (w2): 3600
        m_dns_list (x): (string list with 11 elements):
            156.154.71[.]22
            208.67.220[.]220
            ...
        m_dns_ip (u5): 156.154.71[.]22
        m_smtp_ip (h6): 98.136.216[.]26
    m_tasks (v):
        m_task (d): (4 elements)
            m_address (e): (string list with 2500 elements):
                mc[redacted]@wanadoo[.]fr
                ma[redacted]@wanadoo[.]fr
                ...
            m_body (r): string (323 bytes):
                Received: from unknown (HELO localhost)
                    (%^C0%^Fnames^%@%^Fdomains^%^%@%^C6%^
                I^%.%^I^%.%^I^%.%^I^%^%)
                by %^A^% with ESMTPA; %^D%^R20-300^%^%
                From: %^V0^%
                To: %^0^%
                Subject: Can you have enjoyment 5 times a night?
                Date: %^D-%^R30-600^%^%                 Unbelievable revealing effect on male health
                http://infinitive.zfjvyfhw[.]ru/
            m_name (g): string (1 bytes): 2             m_address (e): (string list with 2500 elements):
                cric[redacted]@optusnet[.]com.au
                claire_fri[redacted]@health.qld[.]gov.au
                ...
```

Distributed Denial-of-Service Attacks

A relatively unknown feature of Kelihos is the bot's capability to participate in DDoS attacks. To command an attack, the botnet operators simply included a special section with parameters specifying the target, type, and intensity in the tasking messages that are periodically requested by each infected machine. The following is a representation of a parsed example configuration that would instruct the bot to perform HTTP requests against the specified IP address, with 50 simultaneous connections and a delay of one second between cycles.

```
m_ddos_config (jz):
   m_config_id (y): 2017-03-14 13:43:57 GMT (58c7f39d)
    m_attack_list (z5): (string list with 1 elements):
       185.53.168[.]141:80
   m_sockets_count (s): 50
   m_sleep_msec (e): 1000
   m_flags (w): 01
```

DDoS commands in the Kelihos botnet were rare. We observed an attack in November 2016, followed by several months without the feature being used before more attacks were launched between March 14 and 17, 2017. The first of these new attacks started on March 14, 2017 at 13:43:57 GMT and targeted the host 185.53.168[.]141. According to historic DNS information, this IP address was at that time associated with the website `ikra[.]top`, a Russian internet portal for running advertisement campaigns on websites. This attack configuration was last

observed on March 15, 2017 at 18:44:54 GMT, immediately followed by an attack against 104.31.253[.]10, and later against other IP addresses in the 104.16.0[.]0/12 network range. This network is operated by Cloudflare, a provider of DDoS protection services. The previously targeted advertisement service appears to have moved to Cloudflare following the initial attack, which may be the reason for the reconfigured targeting.

A second attack that targeted the host 154.46.32[.]129 started on March 14, 2017 at 14:44:42 GMT. This IP address was associated with the Russian Bitcoin exchange online service utbs[.]ws. After five hours, the target was reconfigured to 151.139.244[.]11, an IP address operated by DDoS protection service provider *StackPath*, headquartered in Texas, that took over the hosting of the exchanger's website. A short-term reconfiguration to 107.154.147[.]104 — an address operated by protection service provider *Incapsula* — cannot definitively be associated with this attack but it is likely related.

The botnet operators' motives behind these attacks are unclear. Kelihos was primarily used for the distribution of spam emails. Of note, both targeted sites offered Russian-language online services that were potentially attractive for criminals acting out of Russia. It is possible that the operator of Kelihos was engaged in a business relationship with the targets and launched retaliatory attacks after non-beneficial deals, or in order to inhibit a competitor. Another possible explanation is that these attacks were an attempted vehicle for extortion.

Click Fraud

Levashov constantly sought new ways to monetize infections. To further increase criminal revenue, a feature was added to generate fake clicks on websites. This became part of a click-fraud affiliate program that was started at the end of 2013 and operated at the website `sevpod[.]com` . However, for unknown reasons, the click fraud operation was suspended in 2016 and the clicker feature has rarely been used since then. We observed that this feature was use, in this case, for the purpose of what appears to be a DDoS attack. The clicker code was still present in the final build of Kelihos, and job servers still provided a configuration for the module until its demise.

The clicker operates by instantiating an `IWebBrowser2` Object Linking and Embedding (OLE) object to control an Internet Explorer browser in a hidden desktop. The `m_sites_list` field in job tasks specified a list of URLs to visit. The webpage is parsed using the `IHTMLDocument2` interface to extract the links on the page. After extraction, the Kelihos clicker module tries to blend in with normal user behavior by moving the mouse to random locations with varying speeds (between 50-300 pixels per second) before clicking on the site's links — which in the event of a click fraud campaign would include online advertisements. An example of a (blank) clicker configuration is shown below.

```
m_clicker_config (8):
   m_config_id (4): 2017-04-03 18:02:01 GMT (58e28e19)
   m_sites_list (p): (0 elements)
```

IP Filter List

There were a total of four successful takeovers of the Kelihos botnet through peer list poisoning. With the exception of the last, Levashov purchased new infections and recreated the botnet from scratch in each case. To make future disruption efforts more difficult, the author of the Kelihos malware added countermeasures to prevent researchers from tampering with the peer-to-peer network. One such measure is an IP address blacklist that instructs infected systems to prevent communications with any suspicious peers.

```
m_ip_filter_config (34):
   m_config_id (y): 2017-03-31 22:40:09 GMT (58dedac9)
    m_block_loopback_ip (r): 1
   m_hosts (e): (string list with 1403 elements):
       0.0.0[.]0
       0.0.0[.]1
       0.0.0[.]2
       0.0.0[.]3
       127.0.0[.]1
       87.226.16[.]42
       ...
```

Fast-Flux DNS Hosting

One of the most important features of Kelihos, beyond its spam abilities, was its ability to serve as a fast-flux hosting service. To facilitate this service, the botnet implemented a technique known as double-flux DNS where both DNS A records, the one for the nameserver and the one for the domain itself, point to infected machines. These entries used a time-to-live (TTL) value of 0 to prevent caching. This could be observed by performing a DNS query for the domain `ns1.goloduha[.]info ,` as shown below:

```
; <<>> DiG 9.9.5 <<>> ns1.goloduha.info
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53362
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns1.goloduha.info. IN A ;; ANSWER SECTION:
Ns1.goloduha.info. 0 IN A 27.147.125[.]109
```

Similarly, performing a DNS request for the domain goloduha[.]info produced the following output:

```
; <<>> DiG 9.9.5-3ubuntu0.8-Ubuntu <<>> goloduha.info
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43375
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;goloduha.info. IN A
;; ANSWER SECTION:
goloduha.info. 0 IN A 112.185.217[.]57
```

A parsed configuration that instructs a subset of router nodes to participate in the fast-flux service is shown below. The `m_domains` field specifies a list of domains for which queries should be answered, and the `m_hosts` field provides a mapping for these domains to a list of router node IP addresses. The code in the malware responsible for checking queried domains against the list uses regular expression pattern matching and makes sure that arbitrary subdomains resolve as well. Any HTTP request for a website hosted on the fast-flux network was forwarded upstream through a router node.

```
m_dns_config (j):
   m_config_id (5): 2017-03-30 18:42:40 GMT (58dd51a0)
    m_domains (yy): (string list with 9 elements):
       betaler[.]com
       greystoneexpress[.]com
       gorodkoff[.]com
       combach[.]com
       shoponlinesoft[.]com
       goloduha[.]info
       zavodchikshop[.]com
       ykxitfaf[.]ru                          m_hosts (d): (string list with 29 elements):
       101.96.39[.]79
       112.185.217[.]57
       ...
   m_domain_ips (a): (0 elements)
```

SOCKS Proxy Service

Kelihos utilized two primary methods to distribute spam: via direct SMTP using its built-in spam engine, and by leveraging stolen email credentials to authenticate and spam through legitimate mail servers. In order to mask the origin of a spam run, Kelihos hosted a SOCKS5 proxy service on router nodes. Access to this proxy network was rented out to other criminal groups and was restricted by providing a list of allowed IP addresses in job tasks. An example configuration is shown below:

```
m_socks_config (d):
   m_config_id (s): 2017-02-12 14:21:44 GMT (58a06f78)
   m_allowed_ip (f): (string list with 1377 elements):
       102.118.103[.]102
       102.121.102[.]117
       ...
```

Network Packet Capture

Kelihos utilized the *WinPcap* library to snoop on a host's network traffic, searching for the plaintext protocols HTTP, FTP, POP, and SMTP. Captured credentials were exfiltrated to the job servers via router nodes.

FTP Account Harvesting

Kelihos has built-in support for stealing credentials stored in password managers for 51 different programs. In the past, ZOMBIE SPIDER has used these stolen credentials to host malicious content on the respective servers. The following list shows all programs targeted by the Kelihos FTP credential harvester:

- `32bit FTP`
- `3DFTP`
- `ALFTP`
- `BitKinex`
- `Blaze`
- `BulletProof`
- `Classic FTP`
- `CoffeeCup`
- `Core FTP`

- `CuteFTP`
- `CyberDuck`
- `Deluxe`
- `Directory Opus`
- `ExpanDrive`
- `FAR Manager FTP`
- `FFFTP`
- `Filezilla`
- `FlashFXP`
- `FreeFTP/DirectFTP`
- `Frigate3`
- `FTP Control`
- `FTP Explorer`
- `FTPGetter`
- `FTPRush`
- `Global Downloader`
- `IE`
- `LeapFTP`
- `Leech`
- `Linas`
- `MyFTP`
- `NetDrive`
- `Netfile`
- `Nexus`
- `Notepad++`
- `NovaFTP`
- `Putty`
- `Robo`
- `SecureFX`
- `Sherrod`
- `SmartFTP`
- `SoftX FTP Client`
- `Staff`
- `TFTPInfo`
- `TurboFTP`
- `WebDrive`
- `WebSitePublisher`
- `Windows/Total Commander`
- `WinSCP`
- `WISE`
- `Wisper/Surfer`
- `XFTP`

USB Spreader

Kelihos has built-in support for spreading via removable drives. The spreading process replaces directories on removable drives with a `.lnk` file that, when accessed, executes a Kelihos binary before displaying the expected directory contents. The executable filename was randomly chosen from the list below, appended with an `.exe` suffix. The USB spreader could be enabled or disabled based on the variable `m_use_hello_friends` in the `m_general_settings` section of job messages. The malware executable's file attributes were set to *hidden* so that it will not appear in a standard directory listing.

- `click`
- `game`
- `hentai`
- `install`
- `installer`
- `password`
- `porn`
- `run`
- `screensaver`
- `setup`

Bitcoin Mining and Theft

Prior versions of Kelihos had the ability to mine Bitcoins; however, the mining process has become too computationally expensive for modern CPUs. As a result, the malware has switched to stealing wallets from infected systems instead. Kelihos specifically searches for the following Bitcoin wallet files on a victim's system and exfiltrates them if they exist:

- `%APPDATA%\Bitcoin\wallet.dat`
- `%APPDATA%\Roaming\Bitcoin\wallet.dat`

Attribution

The main threat actor behind ZOMBIE SPIDER, CrowdStrike's name for the Kelihos operation, is Peter Yuryevich LEVASHOV. He used the moniker *Peter Severa* or *Severa*. Levashov was a member of several Russian underground forums where he advertised his products and voiced his opinions. The avatar used by Levashov was a dragon-like character shown in Figure 3.

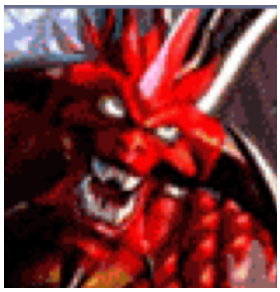


Figure 3. Avatar Used by Severa

Shortly after his arrest, forum accounts associated with the user Severa were banned, removing his contact information and historical posts. Prior to Levashov's arrest, the actor was routinely communicating in the underground marketplace under the alias Severa to advertise his botnet and spam services, his click-fraud affiliate program (SevPod), his fake antivirus program (SevAntiVir) and access to criminal infrastructure.

Severa regularly advertised his spam services on criminal forums ranging in price from $200-500 USD, depending on the service required. Services varied — from pharmaceutical advertising to employment lures to Trojan distribution. In 2014, during a period in which several countries imposed sanctions against Russia, Severa offered customers a discount on his spam service for one month, if it was used against countries that played a part in setting the sanctions (for example, Ukraine or countries in the European Union or North America).

Possible Ties to the Russian Government?

There has been speculation that some criminal threat actors, including Severa, have had ties to the Russian government[9] [10]. In criminal underground forums, members discussed Severa having connections to the Russian government, such as the Federal Security Service (FSB), possibly due to his frequent displays of patriotism for Russia. There is no clear evidence that these claims were legitimate, but Severa played into these rumors with a forum post on April Fools' Day in 2013. In this post, Severa discussed an offer that he allegedly received from the FSB to lead a team in their Center for Information Security, in a new division called the OSBIB (Separate Special Battalion of Information Security). He claimed that he was ordered to hire the first 100 members of this team (out of the 500 required within the first year) and that he had been given significant latitude in the recruitment process. Severa described the intent of this new department as protecting Russia from electronic threats and providing a reactive response if required. The post instructed interested applicants to submit a resume and specifically requested details about illegal hacking activities and botnet development. Severa also remarked that if an applicant conducted criminal activity but was hired into the program, they would be given full amnesty.

The primary requirements for applicants was that they have Russian citizenship, be between 18 and 45 years of age, have a strong knowledge of computer security, work well in teams, and possess excellent problem-solving skills. Applicants with a higher education in a technical field as well as those with past military experience were preferred. Severa also stated that those who were successful in the interview process would be subject to a trial period and would receive official officer ranks thereafter. The post explains that the office would be based in Moscow and those employed would receive a full FSB benefits package, including family accommodation, and a salary starting from 150,000 Rubles per month. Severa concluded the post with a message of Russian solidarity, stating that it was time to pay back the motherland.

Although this post was written on April Fools' Day (Severa had also written a post on the prior April Fools' Day about working for Microsoft), there is a possibility that the timing was intentional for plausible deniability.

The most concrete evidence that Levashov may have had a relationship with the Russian government is from when he appeared in a Spanish court in September 2017 to fight extradition to the United States. During the trial, Levashov stated that he had worked as a military officer for Russian president Vladimir Putin's United Russia party for the last ten years, gathering information on the opposition[11]. However, his claim was later denied by United Russia.

The Author of the Kelihos Malware

While Levashov was the operator of the Kelihos botnet, he likely did not write the malware. At least one of the actors responsible for authoring the malware appears to be a Russian individual named *Andrey SABELNIKOV*. There are several strong links associated with Sabelnikov, including his Github page[8] shown in Figure 4. The project, known as *epee*, contains a significant portion of the non-malicious parts of Kelihos, including a custom and complex serialization library, network functions, registry tools, hashing routines, and an SMTP client.

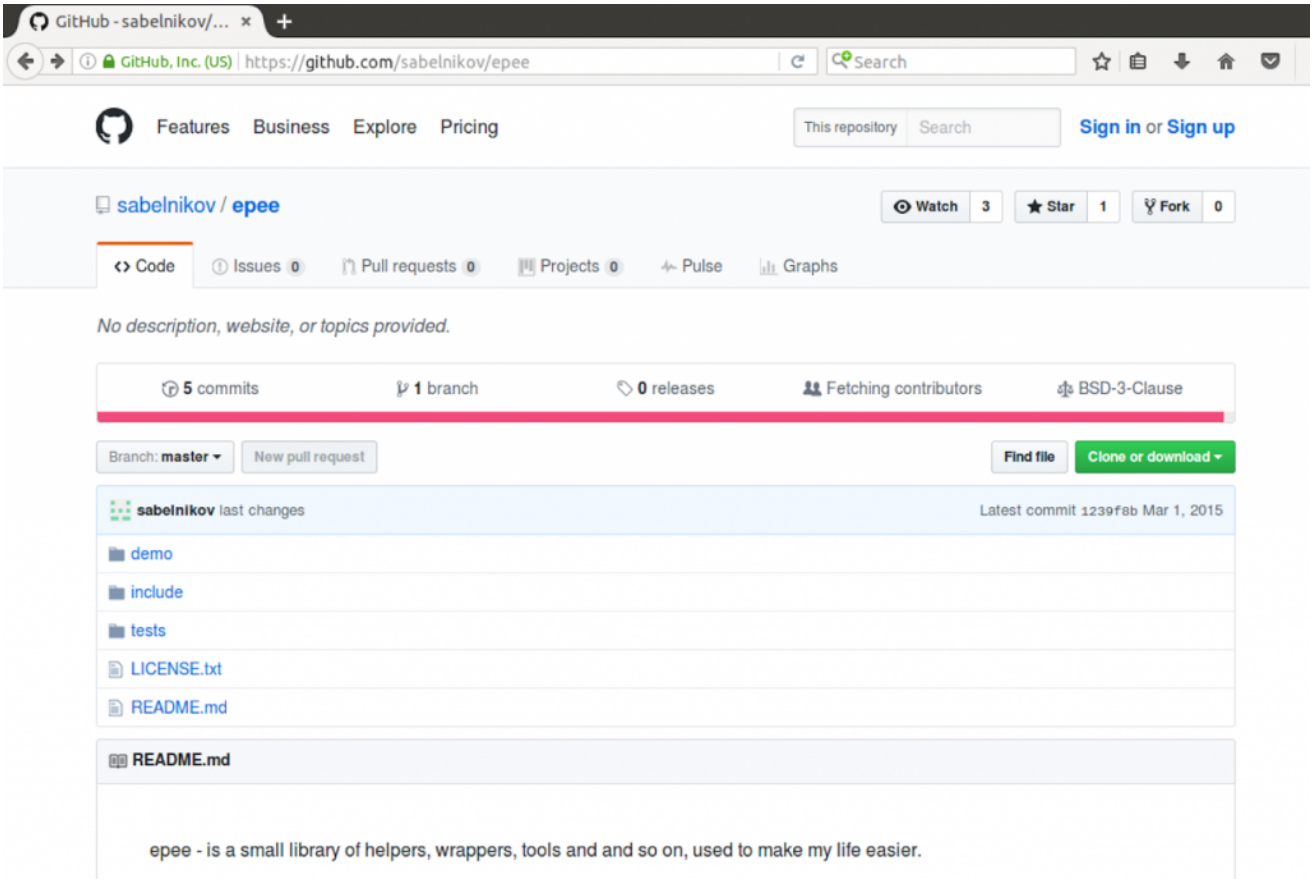


Figure 4. Screenshot of Sabelnikov's Public GitHub Page

Outlook

With the Kelihos spam botnet no longer in operation and Levashov behind bars, multiple criminal operators turned to different spam botnets to distribute their crimeware. Spam botnets such as CraP2P (often wrongly identified as Necurs, which is a for-sale rootkit that is used by multiple botnets) and Cutwail are viable replacement options. CraP2P has frequently been used to distribute other malware such as Locky and Dridex, but also supported large scale spam campaigns for dating advertisement and pump-and-dump scams after the demise of Kelihos. Both these spam campaign types were previously distributed by the Kelihos spam botnet, indicating that respective ZOMBIE SPIDER customers may have switched services.

One spam alternative was offered by a criminal actor who is thought to have been one of ZOMBIE SPIDER's main competitors, having had previous disagreements on who provides better services. This individual reminded users of his spam services in a forum thread about Severa's arrest, posted just three days later.

Appendix

Although the Kelihos botnet is now inactive and no longer under control by ZOMBIE SPIDER, there may be some infections left. In order to help identifying them and cleaning them up, we share the YARA rule below that looks for the three RSA keys mentioned above. Since Kelihos executables are packed with various packers, this rule is unable to detect variants on disk. Instead, it should be used for scanning the memory of running processes.

```
rule kelihos_e : kelihos commodity spambot {
  meta:
      copyright = "CrowdStrike, Inc"
      description = "Kelihos.E embedded RSA keys"
      version = "1.0"
      last_modified = "2018-11-29"
      malware_family = "Kelihos.E"
      in_the_wild = true     strings:
```

```
        $rsakey_extinfo =
"MIIBCAKCAQEArhNkAqO5rfZkXRlmtrZQ4lB0HDPCF9pROK0upgPxKamx7W8mY7GBe3Qk6npYNxHNtV6DN1g+EoSQaMfhpxxlcvMCnvuivJdLN6oQg7UWfqx
        $rsakey_jobinfo =
"MIIBCAKCAQEAn5+cs80qt/4pslfUwTspXxTxVzmk0f9Oxt8on/jyQiuIG/oAhvefsYaDX/xivlvft34T0PhF/8/oAuXCfH4KPJ+GYFLe1hFR7EVdPfVKPRc
        $rsakey_routers =
"MIIBCAKCAQEAqQ8pkPATx8TUt7IaMWXcUwGpkZKmyrHyZj4Asf0f/gXi/FjisO91yNEbuG0ilVNQg+Y4jaycxp/o+iEoEF9CmozwP5F8I9UclBnopTpcHoD
    condition:
        all of them
}
```

Footnotes

1. https://blogs.technet.microsoft.com/microsoft_blog/2010/02/24/cracking-down-on-botnets/
2. https://securelist.com/botnet-shutdown-success-story-how-kaspersky-lab-disabled-the-hluxkelihos-botnet-15/31058/
3. https://www.crowdstrike.com/blog/p2p-botnet-kelihosb-100000-nodes-sinkholed/
4. https://www.crowdstrike.com/blog/peer-peer-poisoning-attack-against-kelihosc-botnet/
5. https://www.justice.gov/opa/pr/justice-department-announces-actions-dismantle-kelihos-botnet-0
6. https://krebsonsecurity.com/tag/peter-severa/
7. https://www.justice.gov/opa/pr/russian-national-who-operated-kelihos-botnet-pleads-guilty-fraud-conspiracy-computer-crime
8. https://github.com/sabelnikov/epee
9. https://www.nytimes.com/2017/03/12/world/europe/russia-hacker-evgeniy-bogachev.html
10. https://www.forbes.com/sites/thomasbrewster/2017/03/20/alexsey-belan-yahoo-fbi-hacker-allegations/
11. https://www.bloomberg.com/news/articles/2018-09-12/russion-who-ran-kelihos-botnet-pleads-guilty-in-connecticut

Learn More:

For more information on how to incorporate intelligence on threat actors like ZOMBIE SPIDER into your security strategy, please visit the Falcon Intelligence product page.
Download the CrowdStrike 2020 Global Threat Report.
Learn more about CrowdStrike's next-gen AV solution, by visiting the Falcon Prevent product page.
Test Falcon Prevent for yourself with a free 15-day trial today.