

**Lessons from Virus Developers:
The Beagle Worm History Through April 24, 2004
Jason Gordon**

Introduction

This paper presents the technical achievement of the Beagle worm as a warning of things to come for security administrators. It does not intend to be an exhaustive technical guide to discovering and removing the worm, nor does it list each detail of how the code works. Rather, the account provided is intended for virus researchers and security professionals as a study in how worm authors improve their products. With this information, it will be possible for those responsible for system integrity to better tune their own tools, policies, and predictions for where a piece of malicious code may attempt to attack.

Note: Virus research is often hampered by the use of different naming conventions and by the nature of virus collection. The Symantec nomenclature was adopted for this paper to maintain consistency and to provide a starting point for anyone wishing to complete additional research (as well as a personal preference for “Beagle” over “Bagle”). This convention is applied in all cases where possible. There will likely be new variants of the Beagle worm released, and lessons learned, after the date above; this paper provides information on those discovered up to that point.

Overview of the Beagle Mass Mailer

Simply stated, mass mailer worms infect computers and attempt to spread themselves via a large number of email messages. Early 2004 witnessed a tremendous boom in mass mailer activity, in terms of both number of worms/variants and also the worms’ infection success. Three mass mail worms, MyDoom, Netsky, and Beagle, accounted for more virus activity in two months than the sum of malicious code did in all of the previous year.¹ Picking up where mail-borne worms such as Sober, MiMail, and Klez left off in 2003, the new wave of mailers refined the art of conning users into opening unrequested attachments. In addition, these worms tested and honed a few new technical tools to aid infection speed and damage.

The Beagle (aka Bagle) worm utilizes its own SMTP engine (in most variants) to send messages to each email address lifted from an infected machine. The worm is named for the executable it originally created upon infection, “bbeagle.exe.”² In early versions, a copy of the worm (crafted during the installation routine) is attached to an email message created in the infected machine’s RAM and sent via the worm’s own SMTP engine. Beagle is equipped with its own MIME-encoding functionality.³

Beagle is an interesting study as it presents new vulnerabilities to the security administrators of the world. Infection and propagation is initiated by having a user open an attachment or simply open an email message. The success of Beagle is certainly grounded in this simplicity. However, as will be seen below, the strength of the worm is built upon a few remarkable technical achievements.

Evolution

Arguably the most striking aspect of Beagle is the dedication of the author or authors to refining the code. New pieces are tested, perfected, and then deployed with great forethought as to how to evade antivirus scanners and how to defeat network edge protection devices. It is this “professional” (a term that may prove to be more accurate when/if the author(s) are ever discovered⁴) process that should be the most frightening to security administrators. Many corporations have adopted process improvement models with great success and boosts to efficiency and product quality. In the world of malicious code, these types of gains are made at the expense of computer users.

Beagle.A

The original strain of the worm appeared on January 18, 2004, in the wake of mass mailing successes like MiMail. MiMail, SoBig, and previous mass mailers proved that the need to have a user open an attachment (whether HTML as in the first MiMail code⁵ or as an executable) as a propagation mechanism is not much of a hindrance to their spread. With the exception of addresses with strings “.r1” or “@avp” and three Microsoft domains (Microsoft.com, hotmail.com, & msn.com), the worm targeted any email address found on the local machine.

The worm used a random string of characters as a filename for the viral code, an attachment with an “.exe” extension, and the Windows Calculator icon (and sometimes opened the application, covering what was being installed in the background). Previous mass mailers (such as 2001’s Sircam) used less recognized extensions such as .pif to fool users into opening them. It appeared, however, that the author(s) of Beagle did not intend for this code to be the finished product. The writer(s) included “Test” in the body of the mass email twice, kept the subject line the same (a simple “Hi” which the writer undoubtedly knew would make it easy to filter), and a self-stopping date of January 28 (10 days after the original detection, and presumably the release).⁶

Beagle.A, although not the most damaging mass mailer up to that point (a distinction certainly up for debate but likely given to code such as Nimda or to SoBig, now being tested by Beagle contemporaries MyDoom⁷ and Netsky⁸), did experiment with a number of functions that the author(s) would employ with great success later. First, the “From:” field of the email was modified so that the “sender’s” address mirrored the domain of the recipient. Second, the worm attempted to retrieve additional code from the Internet by way of a hard-coded list of web servers. As was seen later in Beagle.B more definitively, it is likely that this action also allows the author to catalog infected machines. At the time of detection, none of the servers in question actually had the script (1.php) available. Later reports from infected networks indicated that the worm downloaded a Trojan known as Mitglieder⁹ (code that also sends information such as IP address, port, and ID to a website/PHP page and carries email relay functionality with it). It is impossible to tell if the author(s) placed the Trojan on the sites, if another party placed it there, or if the detection of Mitglieder on machines with Beagle was just a coincidence. Additional

Mitglieder information is available in the Appendices. The worm did, however, test another mechanism to place additional code onto an infected machine, a backdoor on TCP port 6777. The simple set of commands available to an attacker connecting to this port includes a command shell, the ability to download files to the local system, and kill the worm process altogether.

The Beagle propagation scheme is outlined in this simple diagram:

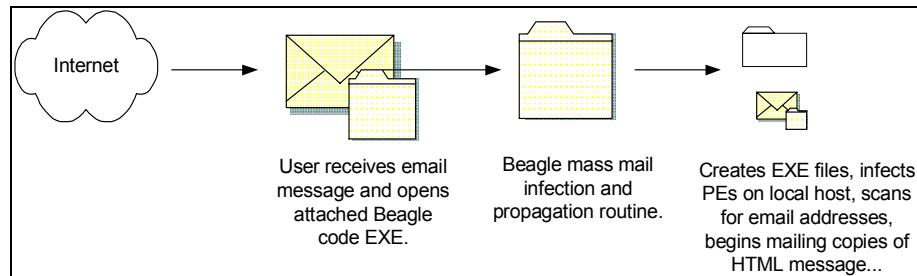


Figure 1: Beagle mass mailer propagation scheme.

Beagle.B

Discovered on February 17, this variant was initially named Alua, Tanx, and Yourid by some anti-virus vendors before the community settled on the fact that the new code was a modification of the Beagle worm. This version terminated itself after February 25. Using a similar email routine, the worm still employed a single subject line, a static body, and a single executable with a random filename and .exe extension. It also opened a listening port, this time TCP port 8866. The backdoor component verified two functions during the installation: an update command (-UPD) and a delete command (-DEL). The delete routine is invoked when the worm detects a date after its termination date. Beagle.B also attempted to open a Windows application, this time the Sound Recorder program.

This version of the worm, however, came with a troubling addition: the ability to generate a random ID value for an infected machine and then send the port it was listening on with that value to four hard-coded sites in the form of an HTTP GET. The request was directed to files named "1.php" and "2.php" on the servers in question. Combined with the remote update and control functions, this provided the author(s) with a catalog of machines to carry out any function, including possibly dispersing new versions of the worm.

This catalog of infected machines could also be used for dedicated attacks. For instance, if the author(s) wanted to launch an attack or reconnaissance effort at Company XYZ, all they would need to do is sort their compromised machine list by IP address and match the appropriate target to the appropriate network. This is all done with simple WHOIS lookups and would work for any commercial, military, or government application. Once the specific machine(s) are identified, they can be singled out for advanced versions of the code or specialized attacks.

Beagle.C

Discovered February 28, 2004, Beagle.C contained many of the same processes found in its predecessors. It opened TCP port 2745 this time, sent information (including a unique ID number) back to one of three pre-established URLs, and propagated via its own SMTP engine. The email was sent with one of 33 different subject lines. The attachment was randomly named and sent with a “.zip” extension as a compressed archive. Inside the .zip archive was the worm, titled README.exe. There was no message body. The worm checked for the same update/delete abilities. The self-termination date for Beagle.C was March 14.

The worm included three new features. The first was a tactic attempted by many pieces of malicious code: disabling security mechanisms on the local host. Beagle.C attempts to stop the processes of a short list of updating services, presumably to prevent new virus signatures from being downloaded to the machine. Additionally, the worm included a DNS server address to use in resolving MX records should a local server be unavailable. A more intricate operation took place as the worm started its infection and propagation routine. This time the SMTP engine was inserted into the address space of “explorer.exe” as a DLL (“onde.exe”). A second file, “doc.exe,” loaded the SMTP engine. Host-based firewalls that filter traffic based on originating process may never have caught the worm’s mass mailing efforts, as they would have appeared to come from “explorer.” Further, this memory persistent action requires a restart of a system to fully remove the worm. The injection method of the code into “explorer.exe” continues through each incarnation of the worm discussed here.

The reliability of the worm’s propagation methods is bolstered in many ways in this variant. Inclusion of a last resort DNS server to resolve mail servers, disabling update services to keep the worm alive longer, and attempting to slip the mass mailing engine into a legitimate Windows process to disguise the worm’s presence all act to keep the code running as long as possible on machines while garnering little attention.

Beagle.D

Discovered shortly after Beagle.C on February 28, Beagle.D is nearly identical to the previous incarnation. The one change is that the mutex created to ensure a single instance of the worm is running at all times is named “iain_m2” in this version as opposed to “imain_mutex” in Beagle.C.

Beagle.E

This version of the worm was also discovered February 28. At this point, the worm included a short text line as the message body, selected from a preconfigured list. It has the same termination date as Beagle.C and .D. All versions of the code up to this point have been compressed with UPX; this variant is PEX compressed and is approximately 2KB larger than Beagle.C when it arrives. When copied to the local disk, Beagle.E used much different file names than previous versions, with the worm being titled

“i1ru74n4.exe” and the SMTP engine “GODO.exe”. This version uses the “imain_mutex” from .C. The worm still sent information back to predetermined servers via port 80. It continues to use a predetermined DNS server as a failsafe.

Beagle.E changed the termination date to March 25, 2004.

Beagle.F

Beagle.F was discovered February 29, 2004 and was the most widely distributed version. The installer, titled “i1ru54n4.exe” (with the SMTP engine being called “go54o.exe” and the loader “ii5nj4.exe”), performed similarly to .E. The worm still attempted to disable auto-update services, open TCP port 2745, send information back to servers (now once again a request to “scr.php” on the web servers, which could certainly be additional malicious code), and harvest email addresses/mass mail itself. The three web server names used, however, had not been seen in the code before. The worm is packed with PEX. It also terminated March 25, 2004.

The size of the worm has grown quite a bit, from 18,007 bytes to a floating size of 22,528 – 24,033 bytes. This changing value is in part due to the insertion of a random value (between 5 and 1,505 bytes of random characters) appended to the copy of the worm used as the attachment to the mass emails. This again hindered detection and removal efforts, as specific sized archives cannot be filtered at mail relays and anti-virus signatures can no longer use file sizes/checksums as a trigger. Moreover, this greatly obfuscates the internal workings of the worm. Without the use of hash values and checksums for identification, subtle changes in the code may not be detected by analysts. Pieces of the code that may appear to be junk may actually be ciphered counters, logs, or any other data the author may be keeping/spreading.¹⁰

The worm now contained a large number of subject lines (45 unique lines) and much lengthier message bodies. The messages (26 unique message bodies) are conversationally toned lines that spoof entries from a social/dating chat service. There are 30 possible names for the attachment, which gives the impression of being a photograph, in almost all cases with a female name. Attachments were now given .exe or .scr extensions before being placed in a .zip archive. The .zip was also given a name from the attachment name list mentioned above.

```
From: [spoofed address selected from infected machine]
Subject: Hi! :-)
I love to dance, read poetry, make people laugh, and hug as many people
a day as i can.
password for archive: [5-digit password]
Attachment: Sara.zip
```

Figure 2: Sample Beagle.F message.

An additional, and incredible, feature of the worm is that it sometimes generates a password-protected .zip file as the attachment to the mass emails. If it does, an additional line will appear in the message body, indicating the password (always a 5 digit number)

that will unlock the file. Encrypting .zip files with this generated password allows the .zip to pass through even very sophisticated anti-virus scanners¹¹, a bit of coding sure to be copied by worm developers in the future. Attempts at passing password-protected malicious code were made with Trojans such as Tofger¹², however, never with the scale and success of the Beagle worm. Beagle creates independent copies of the code, generates a password, encrypts the file, and then distributes it to mail recipients around the world in seconds.

These additions to the code fill out the technical portions of the worm that have been tested over the last incarnations. Instead of place headers, with static subjects and random characters for filenames, the author now boosts the life of the code by exploiting social vulnerabilities. Moreover, shifting the size of the attachment, the subject line, and the message body makes it even more difficult to filter at mail relays. The inclusion of the new subject lines/messages to capitalize on human curiosity represents a new level of evolution for Beagle; most technical propagation vectors have been tested, now the packaging was being perfected.

Finally, the worm took on another propagation vector, spreading through any directory with the string “shar” in the name. It used completely different names when copied to these locations and set a Registry value used to prevent more than one spread in this fashion from any infected machine.

Beagle.G

Discovered February 29, 2004, Beagle.G represented a few minor changes to the code. Beagle.G always sends the archive as a password protected file and appends the message body with the password line mentioned above.

All other functions mirror those in Beagle.F.

Beagle.H

Beagle.H was discovered March 1, 2004. It uses an abbreviated list of subject lines (9) and message bodies (4, all single lines). The password line has different syntax. The attachment is now named with a generic filename (from a list of 14) such as “TextFile” and “Info” and is represented by an icon that looks like a folder. The same backdoor functions exist.

Beagle.I

Beagle.I was discovered March 2, 2004. Although the functionality is the same as .H, the files created on the infected machine are now named “go154o.exe” (SMTP Engine), “i1i5n1j4.exe” (loader DLL), and “i11r54n4.exeopen” (copy of worm for attaching to email).

Beagle.J

Discovered late March 2, 2004, Beagle.J applied a new wrapper around the mass mailer by making the recipients believe the message is from a manager/administrator from within their own domain.

This variant completely overhauled the look of the worm, from a user's perspective. The "From:" field used one of five "senders" (management, administration, staff, noreply, or support) and then the domain name of the recipient. The subject of the email, instead of the laundry list of choices previously used, is now a modest seven choices long, made up of warnings concerning the recipient's email account. The message body also contained a randomly selected warning to the user concerning (appropriately enough) the integrity of their email account and instructs the reader to open the attachment for details/instructions on how to keep their machine/account safe.

Examples of some of the message text:

<p>Your e-mail account will be disabled because of improper using in next three days, if you are still wishing to use it, please, resign your account information.</p> <p>Some of our clients complained about the spam (negative e-mail content) outgoing from your e-mail account. Probably, you have been infected by a proxy-relay Trojan server. In order to keep your computer safe, follow the instructions.</p>

Figure 3: Sample Beagle.J message bodies.

These messages (6 possibilities) are followed by a line indicating that details can be found by reading the attachment. After that note is a closing "The ____ team" where the domain of the recipient is inserted, a URL of "http://www." followed by the domain of the recipient, and finally a signature line (example are: "The Management," "Kind Regards," and "Best Wishes." Messages that arrive with the .zip attachment also had a line of text stating that the file is password protected for "security reasons" and then provided the required password.

The new version of the code also changed the Registry location/values used by the worm. This is likely an attempt to prevent old signatures from detecting/removing the worm successfully. Instead of the cryptic letter/number combinations in the past, the worm now used the name "irun4.exe" for the SMTP engine dropped on the compromised host. Other files (the loader and copy) retain the names of the previous version.

The worm still opened TCP port 2745, attempted to send itself to every address lifted from the infected machine, and copied itself to "shar" directories. The backdoor information was sent to the same servers.

The worm came with a termination date of April 25, 2004. This new termination date comes with significant changes to the look of the worm, which has become a common trait of the code.

Beagle.K

Beagle.K was discovered in the wild on March 3, 2004. The code looked very similar to the last incarnation, with the notable exception that the worm used new names for all of the files it created on an infected machine. Each of the 3 files (the SMTP engine, the loader, and the copy of the code) utilized “winsys.exe” (with the loader named “winsys.exeopen,” and the copy “winsys.exeopenopen”). The Registry value changed as well, pointing to “winsys.exe” with a slightly modified key.

All other features of the worm are identical to those of .J.

Note: Some AV vendors captured variants of the .K packed with ASPack and subsequently cataloged the code as new variant strains.

Beagle.L

Found in the wild after a rash of Netsky variants, Beagle.L was discovered March 9, 2004. The variant itself cannot be considered a worm in the strict sense of the word, as it does not contain a propagation mechanism. Beagle.L created the following files: irun4.exe (copy of code), iinj4.exe (DLL loader used to inject system.exe into explorer.exe address space), and system.exe (DLL that acted as email relay).

In a significant turn, Beagle.L now installed a version of Mitglieder¹² as “system.exe.” This Trojan turns the infected machine into an email relay, listening for instructions on TCP port 11117. Recall that the earliest versions of Mitglieder were discovered as being downloaded by Beagle.A in January 2004. The Trojan downloaded an “exceptions” list and titled it “BAN_LIST.txt.” This list tells the proxy what addresses to ignore when relaying email. The malicious code still attempts to send host information to a web site.

As mentioned above, this variant does not contain the self-propagating functionality of the previous versions. Beagle.L likely spreads via previously infected machines, using that SMTP engine to mass mail copies of the new Trojan or install directly to the compromised machines.

Beagle.M

Discovered on March 12, 2004, Beagle.M is reported as a variant of the .K version by some AV sites. Again, this version of the code had no built-in propagation mechanism; it had to be mass-mailed (presumably from machines previously compromised and cataloged by the virus author(s)). Beagle.M followed the same installation process, creating a copy of the code, a loader, and the Trojan (injected as before into the explorer.exe space).

The Trojan (another version of the Mitglieder code) opened a random port above 2000 and alerted 2 websites to the IP address, open port, and ID of the compromised machine. It also connects to 2 separate websites to download a copy of the list of IP addresses for

the Trojan to ignore. The random port acts as the connection point for remote control and as a mail relay.

The malicious code attempted to kill the same security program processes.

At this point the Beagle worm has officially been labeled simply variants of Mitglieder by many AV vendors, creating some confusion with the later variant names. Beagle resurfaces, however, with the SMTP engine and renewed propagation prowess just 24 hours later.

Beagle.M (@mm)

Some overlap exists based on whether the AV vendor site reported the last incarnation of the code. Trend Micro named this code Bagle.N. Symantec's Deep Sight service made the distinction by adding the "mm" (mass mailer) to name the worm and match what is done on the public Symantec Security Response site. This signaled the reintroduction of the Beagle code as a self-propagating virus, with additional insidious features.

There were six significant changes to the Beagle code and functionality at this point. The number of changes in itself is out of character for the coder in question, up until this point multiple additions to the mechanics of the code (beyond simply swapping cosmetic pieces such as subject lines) in a single instance were rare. The changes were:

- Installation routine now creates 4 files instead of 3
- Password sent as a graphic file
- Removal of Registry keys used by Netsky worm
- Greatly expanding termination date of code
- Use of RAR extension
- Polymorphic EXE file infection

The installation routine created multiple copies of itself: winupd.exe, winupd.exeopen. It also created a copy that was sometimes password protected, winupd.exeopenopen.

The password used to open/decrypt the email is now sent as a BMP file (if winupd.exeopenopen is password protected) called winupd.exeopenopenopen. The password is displayed in email messages by including the image after the "password is:" line, not printing it directly as part of the text. When added to an email, the code is given a random filename once again, from 5 to 9 characters in length. This use of a graphic file is similar to applications in the anti-spam arena, where pictures are used instead of text to ensure a human is reading/responding to requests for accounts, etc. Further, a graphic of a password is more difficult for anti-virus scanner to grab from the email to use in decrypting/scanning the attachment. The use of an anti-spam trick to install a spam relay is not without irony.

In a twist not yet seen by the Beagle code, it now attempted to remove 14 keys/values associated with the Netsky worm (which had been removing values created by Beagle for some time).

Beagle's previous variants had self-termination within a month of release; this version's is December 31, 2005, well over a year and a half after the release into the wild.

The copies of the code sometimes use RAR extensions. RAR, another archive format similar to ZIP, is capable of many encryption/archiving techniques including the ability to archive and split files.

The most striking change is the inclusion of another propagation vector: infection of Portable Executable (PE) files. This addition makes the nature of the code even more difficult to classify; Beagle now has virus-like qualities in the strict sense: it writes itself to EXE files to spread. The worm added to EXEs is encrypted, it is decrypted at the time the EXE is run. The compressed/encrypted version of the code increases infected files' size by 21 KB.

Beagle.M includes the previous spreading mechanisms, mass email (by collecting addresses and using its own SMTP engine with a shell similar to Beagle.J) and the ability to copy itself to directories with "shar" in the name. Once it sends itself, the worm makes an entry to HKEY\SOFTWARE\winupd to prevent duplicate emails to the same address. Future infections also check this key to identify whether Beagle.M has already run on the machine.

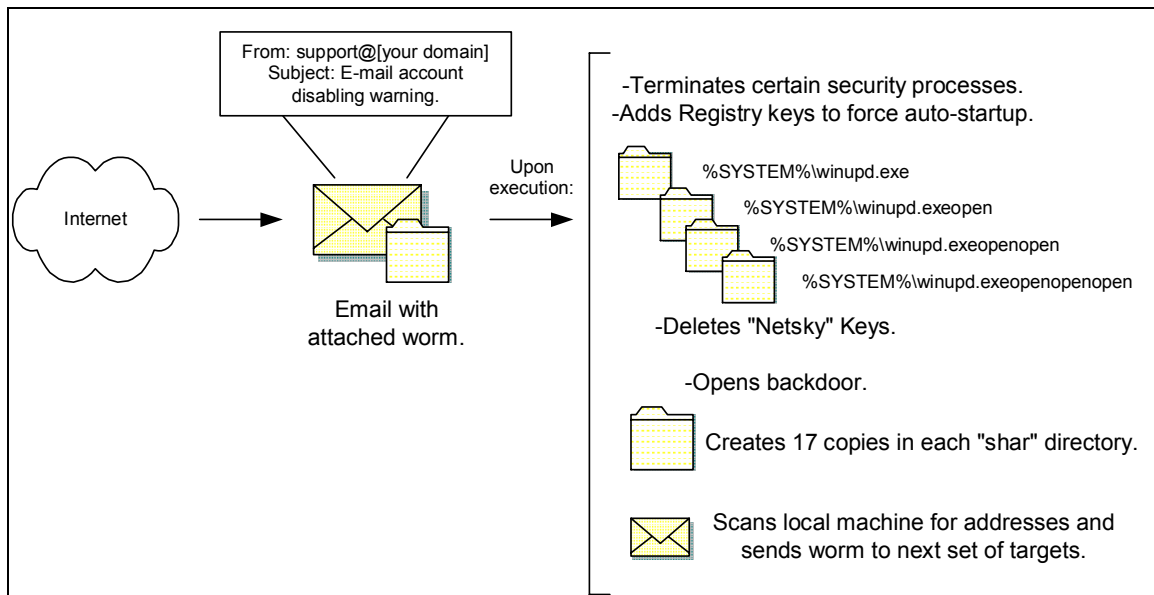


Figure 4: Beagle.M infection components.

The code attempts to stop a greatly expanded list of applications, a list similar to those used by other virii such as Aphex and Agobot. Beagle.M opens port 1220 and may query a DNS server at 217.5.97.137 to resolve MX records. The backdoor allows arbitrary code to be saved to the infected machine with the filename of "iuplda[random characters].exe".

In a departure from the very business-like messages of recent variants, this version also introduces a few randomly appearing strings that reference the film, "The Matrix." Some

emails will contain one of two lines: “Follow the wabbit” or “Find the white rabbit.” Additionally, one of the file names used for peer-to-peer distribution is “Matrix 3 Revolution English Subtitles.exe.” In addition, an ASCII butterfly, preceded by the words, “The white rabbit presents” and “The first and the single Anti-Netsky AntiVirus” appears in the code for the worm (See Appendix, “Hidden Text Timeline” for picture).

The email construction itself has undergone some changes. If the “From:” field is created from an address taken from the local machine, the worm selects from less formal greetings/messages (i.e.: the short “RE: document\see attached” messages). If the “From:” is crafted from the host’s domain, the more formal (i.e.: “email account problems”) messages are sent.

This version of the code makes no call on its own to a web server to download Mitglieder or any other Trojan, nor does the worm attempt to send host information to another server. However, newer versions of Mitglieder continue to appear.

The code is appended with random data, making the file size vary from 20, 485 bytes to 21,985 bytes.

Beagle.N

Discovered March 15, 2004, Beagle.N combines a number of previously employed tactics into a complex propagation scheme. Beagle.N continues to infect PE files, boosting the size of the executables by 44 KB. It still creates a backdoor on TCP port 2556. The worm ends a long list of security/anti-virus programs, kills Netsky infections by targeting their Registry keys. The worm includes the same termination of December 31, 2005 as .M. The files created as part of the installation routine are the same as the last variant; it continues to appear in mailboxes as a PIF, ZIP, or RAR file. The ASCII butterfly still appears in the code.

Beagle.O

Discovered March 18, 2004, Beagle.O is the first of many variants for the day spread purely by file attachments/infection. It creates the four files used to spread the worm with the name “directs” (a name carried into the next iteration of the worm). The code executes an infection routine much like .N. Attachment names are randomly assigned EXE, PIF, ZIP, or RAR extensions (archived appropriately and with passwords). The worm opens TCP port 2556, infects PE files, and spreads via copies to directories with the string “shar” in their name.

Beagle.Q

Up until this point, Beagle exploited only the trust of users. Its email attachments required a recipient to open them. With this variant the author takes the worm in a new direction, exploiting a software bug that allows the worm to spread without attachments, simply by opening an email message. Beagle.Q takes advantage of the Internet Explorer

Object Tag Vulnerability. This flaw allows malicious HTML code to download and execute arbitrary files. Microsoft released two advisories regarding this bug, MS03-032 and then MS03-040¹⁴.

Beagle.Q arrives as an HTML email with familiar subject lines and “From:” fields. The message body appears empty, as the code that downloads the first piece of the installation routine is not visible to the user. This code retrieves an HTA (HTTP Application¹⁵) file that contains just a few lines of HTML and then the VBS (Visual Basic Script) file (dropped as “q.vbs”). This file is downloaded from one of 592 servers hard-coded into the worm utilizing TCP port 81. These servers were previously compromised/deployed by the worm writer(s) to distribute the new code. The HTA/VBS download routine is invisible to the end user, the “window” opened for the transfer is specially coded to be out of the user’s view. The VBS file is then executed. This script downloads another file from the same server that provided the initial code (again on TCP port 81), this time a version of the Beagle.Q executable very similar to the last variant.

The Beagle code is retrieved as a graphic file, meaning that it simply has an extension of .jpeg, .gif, or .bmp. In tests done with the worm by AV vendors and in independent closed testing, the worm never requested anything other than JPEG files. The name of the file is randomly created and encoded into the VBS file along with one of the host servers. As the “graphics” file is downloaded, it is saved as “sm.exe,” which is accomplished with a line in q.vbs. The end of the VBS file executes sm.exe, beginning the infection and propagation routine similar to other Beagle variants.

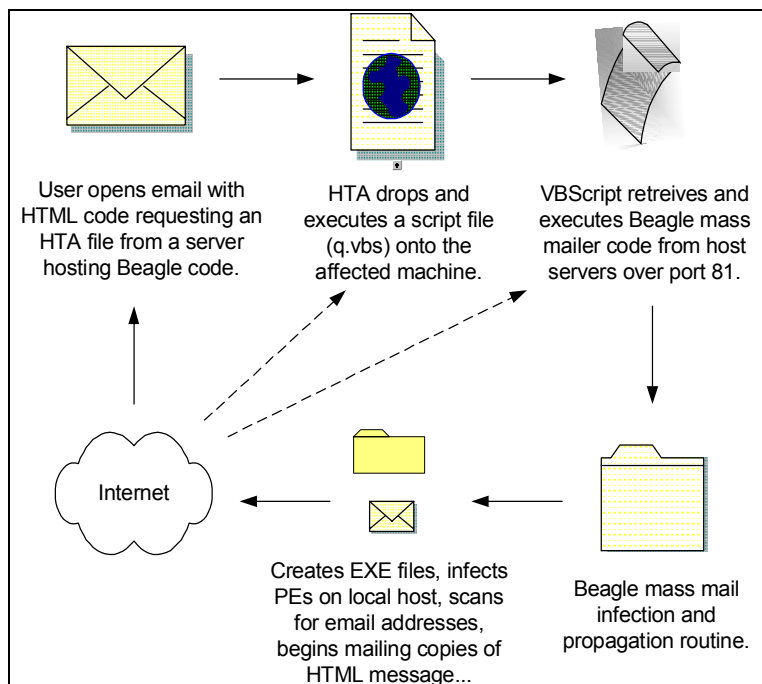


Figure 5: Propagation scheme for Beagle.Q.

The worm keeps a backup DNS server in the code that is used if no DNS server exists for the infected machine. In addition, it drops the same copies of itself into directories named

with the string “shar” as other versions of the worm. Furthermore, it continues the aggressive PE infection of previous Beagle incarnations by adding 26 kilobytes to executables it finds. In addition to opening TCP port 2556 for remote command execution, the worm also opens TCP port 81. Port 81 allows others to download a copy of the worm, much like the hard-coded servers that distribute the code. As of this version, there is no mechanism built into the worm to allow infected boxes to utilize this function for propagation/updates. Undoubtedly this “feature” is being tested for future release.

Beagle.Q scans the entire local machine for addresses. Each address (save those in an exceptions list built into the worm, see Appendix) is sent a copy of the HTML message. Although the means to spread the worm as an attachment (Beagle “Classic” used by the author of this report to describe previous functionality) is present in the code, that portion is unused by this variant. The worm does, however, continue to attempt to stop anti-virus and other security products from running. This version also attempts to remove the Registry keys/values of every NetSky variant up to this point as well. It adds an entry to the Registry for both the automatic startup (using the file it created, “directs.exe”) and to log mass mail activities (kept in HKEY_CURRENT_USER\SOFTWARE\windirects).

If the system date is January 1, 2006 or later, the worm terminates by removing the Registry keys/values it created and exits all Beagle functions.

Detection of the worm has also changed. Many AV vendors broke the worm into its 3 components and issued signatures accordingly¹⁶. The worm is now listed as the Beagle/Bagle email, dropper, and the mass mailer proper.

Beagle.R

Functionally equivalent to Beagle.Q, and released on the same day, this variant only scans the “C:\emails” directory for email addresses to target with copies of the message. It also changed the file names\associated Registry entries to “direct.exe” instead of “directs.exe.” All other details are the same as Beagle.Q.

Beagle.S

In an apparent effort to boost the spread of the worm by hampering signature/identification efforts, Beagle.S, the third variant discovered March 18, 2004, also uses “direct.exe” and scans the single directory, “C:\emails.” It also includes a butterfly picture and text similar to Beagle.M.

Beagle.T

Beagle.T, the fourth and final variant discovered March 18, 2004 returns to the use of “directs.exe.” It also scans the entire local machine, like Beagle.Q. However, Beagle.T does not add the “logging” key in HKEY_CURRENT_USER\SOFTWARE.

Beagle.U

As could be expected, the termination date for .E and .F brought another variant of the worm, Beagle.U, discovered March 26, 2004. This version uses a clock icon for the attachment. It employs FSG 1.33 packing, which makes the code approximately 8 kilobytes (from 37 KB). The worm comes packaged in minimalist style; the email message has no subject, no message body, and a randomly named 8.23-kilobyte attachment that ends with EXE. When executed, the worm drops the file “gigabit.exe” on the local machine and makes the requisite Registry entries to ensure it starts with each boot of the machine and downloads a copy of the worms entitled “a.exe.” It adds a Registry key to track its own activities as well. To disguise the application that is being installed, the worm launches the game MSHEARTS as part of its operation. This recalls early attempts to open the Calculator and Sound Recorder.

This version opens port 4751. The worm also tries to connect to a web server, reporting the address, opened port, and ID of the compromised computer. The worm attempts to send an email to a new target every 5 seconds, as opposed to the blitz of messages sent out previously. This variant is the first to move the termination date ahead of the previous version. Beagle.U exits if the system’s clock is January 1, 2005 or later.

From the stripped-down look of the worm’s shell, it would appear that it is an early test strain. Some reports have suggested this was an initial version of the code that was unreleased until now¹⁷. This version of the worm accounted for a faster propagation (in terms of unique network reports) than many of the previous variants¹⁸. It is possible that previously infected machines were used to distribute Beagle.U especially quickly. However, it is also likely that the MSHEARTS cover worked to prevent users from investigating the attachment further and possibly discovering and cleaning the worm.

Beagle.V

Discovered March 29, 2004, Beagle.V represented just a few changes to the last variant. The version calls a currently unknown application “dreder.exe” upon infection and changes the attachment’s icon to what appears to be a syringe. The attachment is called “game.exe,” which launches the Beagle infection once opened. That executable installs the worm code, open TCP port 4751, and allows for the code to be updated by the same “update” function (-UPD) seen previously. Once a newer version of the worm is successfully placed on the compromised machine the old version is deleted.

Testing of functions such as these would be easy to hide in the myriad of machines that are infected by such worms. The attacker could simply select a few machines (from the catalog generated by infected computers and sent to the author-controlled site), test the update/delete commands, and then consider this phase completed. The updated code would likely be removed via the “delete” string, preventing it from being detected and analyzed by anti-virus vendors. In fact, the code that is tested may look nothing like the Beagle worm familiar to security professionals.

Beagle.W/Mitglieder.F

The name confusion is attributed to the same problem found with the Beagle.M/Mitglieder variant. Without a propagation mechanism, it is not classified as a “worm” or “virus,” and is such classified as a Trojan.

Beagle.W was discovered April 5, 2004. Once executed, it drops a copy of the code with the familiar filename “irun4.exe” (Beagle.J) onto an infected system. That file is loaded with “iinj4.exe.” Both executables are included in the replica of the Trojan, spawned as “system.exe.” Beagle.W opens a backdoor on port 17771. This port allows external devices to relay email through the infected host. The Trojan runs within the explorer.exe space and utilizes a hidden window (“ShellTray_Wnd” to hide itself). It attempts to kill security product processes as in previous versions. The code records its activities in a Registry key (HKEY_CURRENT_USER\Software\DateTime) containing values for the machine ID, process ID, and open port.

Beagle.W downloads a file (saved as, “ban_list.txt”) from one of 16 web servers. The Trojan then attempts to connect to randomly generated IP addresses on port 4751 (the port opened by the two preceding variants, .U and .V). Beagle.W then waits for commands from the previously compromised machines. It uses the DNS server (217.5.97.137) introduced in Beagle.M to resolve addresses for the hard-coded domain names and presumably the MX records of email destinations.

Beagle.X/Mitglieder

Discovered April 7, 2004, Beagle.X represents a newer version of the Mitglieder code. The file/Registry value are named “window.exe.” The port used is 14247. The backdoor/email relay is equivalent to Beagle.W. This version does not appear to attempt to stop security products from running.

The vast array of names used for Beagle.X is evidence of the difficulty in tracking virus code and updates after a number of variants. Where some vendors continue to use the name Beagle/Bagle, some have called the last two versions Mitglieder:

Trend Micro & Symantec	Bagle.X
Sophos	Troj/Bagle.X
Computer Associates	Mitglieder.AC
McAfee/Network Associates	W32/Bagle.X!Proxy
Kaspersky	TrojanProxy.Win32.Mitglieder
F-Secure	W32/Mitglieder.AI

Many vendors had not received samples of the code days after its release. Lack of submissions may be an indication of a small distribution, or that users that are infected do not notice anything strange with their machines (as they would likely have already had a previous version of Beagle installed and active). F-Secure reported that the Trojan was discovered as an attachment to spammed messages.¹⁹

Additional Variants

Over the weeks following the release of Beagle.X, a few other versions of the code were discovered and reported by various AV vendors. This slow and sparse reporting is possibly due to the code being deleted from machines once its functionality is exhausted and relatively small infection rates. Although, as mentioned above, it is also quite likely due to the low probability that a machine that has been infected with a previous version of Beagle for days or weeks has the capability to discover malicious code and submit it to an AV vendor. Beagle.Z was reported by Panda Software (see references for Panda's web site) April 24, 2004. It is another iteration of the Trojan code that uses port 18881 and notifies a new set of web addresses when an installation is successful on a victim machine. The following day (April 25) produced a report on Symantec's Deep Sight for Trojan.Mitglieder.G. Each had components of previous versions and likely used previously compromised systems for installation.

Additional variants will likely follow this list, as will new worms, building on the successes of Beagle.

Discussion

As with all viral code, the discoveries of new variants and possibly new functions will continue. As of this writing, the Beagle worm has shown successful incorporation of the following infection vectors:

- Mass Mailing
- File Sharing Services
- Infection of EXE files
- Software bug exploitation allowing for arbitrary code execution

Furthermore, it has incorporated a number of functions to multiply the potential damage and/or hamper detection and removal:

- Disabling security program update features
- Inserting itself into a legitimate Windows process memory space
- Memory residency
- Use of hard-code DNS address as a failsafe for finding MX records
- Employing a wide array of subject lines and messages
- Extensive use of social engineering tactics, especially within subject/messages
- Inserting random data into the code to change the file size/checksum
- Generating a random filename for attachments with worm code
- Shifting Registry locations and key names/values
- Changing filenames of code loaded on infected machine
- Use of UPX/PEX to slow reverse engineering
- Using modified PEX/packing methods to avoid generic worm detection signatures
- Installation of a backdoor service
- Generating unique identifiers for all compromised hosts
- Relaying IP address, unique identifier, and open port to author-controlled location
- Use of .zip files to bypass many attachment filters settings
- Use of password protected .zip files to bypass virus scanners
- Distribution of Trojan via previously compromised boxes
- Use of compromised boxes to control other compromised machines
- Incorporating host EXE infection
- Exploiting vulnerabilities to install files/updates from rotating Internet hosts
- Opening legitimate applications to cover background infection process
- Use of hidden windows to hide Trojan activity

In addition, the worm's release appears to follow some sound testing procedures, ensuring the technical infrastructure is sound before adding the subject/message selection process to the email, much in the same way SoBig did in 2003²⁰. SoBig relayed information back to a server (via an ICQ address) and downloaded a file containing additional locations for programs to execute. Beagle combines proven propagation vectors with social engineering tactics to overcome the need to have a user open an attachment (admittedly a task that often seems all too easy to many system administrators).

Beagle avoids the current popularity of utilizing IRC channels with worms to feed remote control commands by having the infected machine send all the information necessary for control to an outside machine (all on port 80). Even administrators that block all outbound access to IRC must now consider blocking the addresses used in the worms. As shown above, however, the code can change quickly and be released many times in a single day. Moreover, the worm itself has a built-in update function (the “-UPD” command noted above) that could conceivably be used to change the code on infected machines as it propagates, also hindering removal efforts as the filenames, sizes, etc. may all change with the update.

The use of previously compromised machines (those infected with a more widely distributed version of the mass mailer) to install the Mitglieder Trojan is another simple, but well developed strategy. By using machines that are currently infected with a worm that is at least a few days old, the author(s) can be fairly confident that the user of the compromised machine is not especially vigilant with security updates. This is good for two reasons: the Trojan will be able to relay mail for a significant amount of time before detection, and samples of the undiscovered code are unlikely to reach AV vendors for study.

Moreover, by opening a backdoor without authentication, the author allows other worms or coders to control the machine and the Beagle worm. Worms such as Jeefo²¹ that infect Portable Executables such as Beagle and change the code could wreak havoc on anti-virus efforts, changing the code before, after, or during the propagation. These worms would act outside of built-in update features such as those in Beagle or IRC-controlled worms. Beagle itself began including such features with version .M, adding the ability to infect PE files on compromised machines.

An astounding feature of this worm is the use of protected .zip files that will run through most anti-virus scanners. Supplying the generated password is a function that will undoubtedly be copied by worms in the future, and will force AV vendors to come up with new mechanisms for scanning network traffic. At its heart, however, Beagle doesn't require the technical conjecture presented above. The worm spreads because of a completely non-technical problem, users opening unrequested attachments. In an attempt to block this worm, system administrators are more than willing to jettison certain levels of functionality and service within their respective networks. Everything from restricting attachments to blocking all email has been attempted to stop viral code from infecting workstations.

It should also be noted that later versions of the worm contained non-displayed text (hidden in the Visual C code for the worm) directed at writers of “competing” mass mailer Netsky (Visual C++ v6). Netsky itself contained lines in later iterations pointed at the author(s) of Beagle. The Netsky worm (Netsky.F and later) attempted to remove the Beagle infection (an approach that would work on Beagle variants .A-.I) on a machine by removing Registry entries. This may explain the high number of variants as possibly artificial (as opposed to a natural evolution of the code to improve its power/speed),

intended to force reverse engineering so that these messages would be found or simply as new vehicles for this battle of derogatory lines.

It has been suggested that the creator(s) of Beagle are tied to the spamming industry that has grown exponentially over the last few years. Beagle, it is argued, could be used to retrieve valid email addresses as well as relay messages on behalf of spammers. The worm is certainly well defined for such a task. Although not seen by the current versions of the code, it is certainly within its scope to transmit the email addresses it uses to an outside agency. Furthermore, it is clearly an efficient mass-mailing tool; changing the contents of the message is an easy task—simply update the code by way of the built-in “update” command. Infected boxes are already relays for email, and since the compromised machines have been “catalogued” by a unique identifier sent to the attacker’s servers, controlling an army of unknowing relays would be a minimal challenge. If there is truly a financial incentive behind code such as Beagle there will be no end to the technical innovation or to the number of these worms released onto the Internet.

Beagle’s Functionality in Brief

Below is a consolidation of the functions tested and incorporated with each version of Beagle. Next to each variant are some of the prominent characteristics of each as well as a descriptive tag for the function. The tags are not exclusive; that is, a Base Function or Social Engineering item may also be considered to hinder detection or extend the code’s life. These are just guides, identifying the main reason one may include the module in a mass mailing worm. With this skeleton it is easier to see how pieces of the code were deployed into “production.”

.A	EXE attachment Opened backdoor Downloaded additional code	Base Function Base Function Base Function
.B	Generated Unique ID Submitted ID/port/address info to author Tested remote control abilities Changed backdoor port	Enhanced control Enhanced control Improved reliability Base Function
.C	Added 33 subject lines Disabled security products DNS server added Injected into explorer.exe	Social Engineering Hampered Detection Improved reliability Hampered Detection
.D	Changed mutex name	Hampered Detection
.E	Added text line Changed Compression mechanism Changed filenames/Registry entries	Social Engineering Hampered Detection Hampered Detection

.F	Inserts random “junk” data Eliminates use of hash/checksums Large shell changes – subjects, attachments, etc. Encrypted payload occasionally Added infection vector – “shar”	Hampered Detection Hampered Detection Social Engineering Hampered Detection Extended Life/Reach
.G	Always sends encrypted payload	Extended Life/Reach Hampered Detection
.H	Changed shell – icon different	Extended Life/Reach
.I	Changed filenames	Extended Life/Reach
.J	Completely revamped shell	Social Engineering Extended Life/Reach
.K	New filenames/Reg values	Hampered Detection
.L	Installs trojan - leverages tool: Mitglieder Utilizing pre-built machine army to disperse	Base Function Base Function
.M	Acts solely as Trojan – changes character	Extends Life/Reach
.M(mm)	Changed install routine EXE infection Added compression – RAR Password as graphic	Hampered Detection Base Function Base Function Hampered Detection
.N	File size increased	Hampered Detection
.O	Changed filenames/Registry entries	Hampered Detection
.Q	New vector – only require opening message Modular infection sequence	Base Function Base Function Hampered Detection
.R-.T	Changes filenames, etc.	Extends Life/Reach
.U-.V	No subjects, messages-covers with legitimate app.	Hampered Detection
.W-.X	Hidden Trojan Email relay Updates/Commands from compromised hosts	Hampered Detection Base Function Base/Detection

What is Gained Reading a Worm's History?

Personal interest aside, investigating the development patterns of other worms, especially those with a similar propagation mechanism and payload, can be extremely helpful in predicting the path of new code. For example, by giving SoBig just a cursory overview, one can see a progression similar to Beagle's²². Each worm used its SMTP engine, process names and Registry Keys that changed with each variant, and termination dates to halt propagation as new variants were released. Moreover, both pieces of code attempt to install proxies on the infected host making it possible to relay email anonymously.

January 2003	SoBig.A	Independent SMTP Engine Lifts email addresses through system scan Downloads Additional Code from Internet Location Attachment is PIF Attempts very specific share replication Reports of Trojan (Lala) being downloaded
May 18, 2003	SoBig.B	Set short-term termination date (May 31) Changed Shell ("From:" field, mutex, etc.) Logs Mass Mail recipients Identified by various names by AV vendors
May 31, 2003	SoBig.C	Hard-coded mail servers Adds network share propagation vector Set short-term termination date (June 8) Attachment now SCR or PIF Downloaded additional code/updates
June 18, 2003	SoBig.D	"From:" field from those lifted or <u>admin@</u> Expanded subject/attachment selections Set short-term termination date (July 2) Opened backdoor/connected to servers (NTP)
June 25, 2003	SoBig.E	"From:" is spoof, "support," or logged on user Expanded subject selections again Use of ZIP to enclose PIF file Set short-term termination date (July 14)
August 18, 2003	SoBig.F	Communicates with external server Set short-term termination date (Sept 10) Installation of WinGate Proxy/Lala Trojan Spoofed sender as above Removed open UDP ports\backdoor Retrieves update information from master server Attempts peer\file sharing propagation

This truncated history of SoBig points out a few rather important steps in its development. Whether or not malicious code like this proves to be the work of paid developers, the work and processes can certainly be defended as “professional quality.” Much like Beagle, which began one year after SoBig first appeared, tested various components with termination dates and later combined them with more elaborate exteriors (subject, From: fields, attachment names, etc.) and additional propagation vectors for maximum distribution. Each signaled its use of backdoors\relays with early attempts to open channels and retrieve code from the Internet. The techniques used quite successfully by SoBig were repeated by Beagle, and will likely be used by worms in the future. Other virus writers will likely adopt the tricks introduced by Beagle. Studying the reasons for Beagle’s success can help security administrators defend against these “professionally” crafted worms. Policy decisions can be made now regarding the acceptance of email attachment blocks, especially when known worms are spreading quickly. Noting the combination of attacks used by worms can make their presence easier to detect; there is not a reliance on a single vector or single symptom. IDS sensors can be configured to watch for telltale signs of mass mailers, IRC backdoors, etc. Worm writers diligently adhere to “go with what works;” the past success of Beagle will provide tools to build upon for years.

Worms now infect, install additional code unrelated to the propagation vector, delete the original code, and then pass control of workstations to the virus authors with remarkable swiftness. One tool that will help improve these programs is the same as on the traditional side of software development: process improvement.

Security Planning Lessons

Studying well-crafted worms such as Beagle can yield a number of tips on how to effectively fight viruses of all kinds. Although nothing like the formal and well-defined models such as SEI’s CMMI²³, this section does offer a few points of improvement for systems administrators charged with protecting data from malicious code. It should be evident from the short life of Beagle (relative to code like Melissa and Code Red that still appear on the Internet) that the changes to a worm’s propagation vectors can render specific tactics and virus signatures worthless very quickly. Many plans revolve around only keeping anti-virus software up to date²⁴. Although this a very effective plan for most desktops, one can see by studying Beagle that the potential for a virus attack before a new signature is deployed is quite great. A strong response infrastructure is required to mitigate threats. This infrastructure includes all decision makers required to apply restrictions to incoming traffic. Basic improvement plans with respect to viral mitigation include:

- Continuous study of virus propagation vectors & software vulnerabilities
- Evaluation of detection tools and signatures
- Review of internal policies (as they relate to how traffic flows into the organization)
- Evaluation of response plans (and all tools available to control traffic flow)

Worm writers are certainly studying vulnerabilities and how to quickly exploit them. Those that come with an advisory statement ending in “execute arbitrary code remotely” are always worthy of patching. Detection mechanisms should be reviewed for how well they can identify abnormal events. IDS signatures that detect mass mailers would be effective at catching Beagle, Netsky, MyDoom, etc. So many worms use the mass mailer vector; this type of signature is a necessity for network threat detection. However, this type of detection mechanism is only possible if there is a tight asset management (knowing what and where legitimate mail servers are) process in place within the enterprise. When new attachment formats (such as encrypted zip archives) are employed, it is critical that decision makers be ready to block the file types at the mail relays. Again, this requires a pre-existing control over the types of mail that are allowed into the network. There are generic tools built into most every network device that can help. These include access list creation, routing policies, mail blocks, DNS dead listing, manual IDS signature creation, firewalling, etc. All of these tools are not necessary to combat malicious code. However, a strategy will be limited to what tactics are available. A sample inventory of these tools preceding the Beagle attacks would have allowed an administrator to take the following actions:

<u>Action</u>	<u>Versions Mitigated</u>
Policy/ Tools to Block Certain Attachments	Beagle.A-V
Block DNS server (hard coded address)	Beagle.C-X
Dead list “registration” sites	Beagle.B-V
Block “From:” fields and/or subjects	Beagle.A, J-K, U-V
Detect/Investigate backdoor port use	Beagle.A-T
Block download sites	Beagle.Q-T
Filter HTA/ActiveX	Beagle.Q-T
Filter Port 81	Beagle.Q-T
Filter Trojan Ports	Beagle.B-X

Mass mailer worms are easy to underestimate and dismiss as annoyances that simply degrade network performance. However, the capability of these simple applications to catalog resources, greatly improve attacker reconnaissance, steal files, and to silently test new exploits on production networks requires attention.

To battle modern worms effectively and completely, information assurance administrators will have to have an understanding of all network facets. Practically, this means having the ability to change detection and rejection systems (IDS, firewalls, router ACLs, AV software, email gateways, etc.) quickly based on pre-established security policies. The lesson of worms like Beagle is that virus writers have great control over their software’s functionality. These authors are becoming more skillful at evading tools that are designed to catch their products. They are improving the quality of their releases in controlled development cycles-and learning from each version’s successes and failures. To effectively fight malicious code, security professionals must learn just as quickly.

Additional Information for the Curious-Order of events for generic (early variant) Beagle infection:

Unpacks 3 files to the local machine (SMTP engine, loader, copy of virus for attachments)

Creates Registry entries to ensure worm runs at each startup

Opens backdoor port

Sends GET to specified web servers

Attempts to halt specific security update services

Scans local disk for email addresses

Transmits crafted email/attachment to each address (with exceptions listed below)

Copies worm to directories with string "shar" in name

-Email Address Exceptions

The Beagle worm disregards addresses with the following strings:

Note-Beagle.A uniquely ignored .rl

@hotmail.com

@msn.com

@microsoft

@avp

After Beagle.A, the worm also ignores:

noreply

local

root@

postmaster@

By Beagle.Q and its variants the list expanded to also include:

@foo

@iana

@messagelab

abuse

admin

anyone@

bsd

bugs@

cafee

certific

contract@

f-secur

feste

free-av

gold-certs@

google

help@

icrosoft

info@

kasp
linux
listserv
nobody@
noone@
ntivi
panda
pgp
rating@
samples
sopho
spam
support
unix
winrar
winzip

Beagle.U and .V cut all but two of the exceptions, add carried the short list of:

@avp.
@microsoft

-Ports used by Beagle backdoors

Ports 6777, 4751, and 8866 are all unassigned by IANA.

Port 1220 is registered to Apple's QT Server Admin.

Port 2745 is registered with IANA to URBISNET.

Traffic spikes on this port can be monitored at the Internet Storm Center site:

http://isc.incidents.org/port_details.html?port=2745

Note: Agobot.HM scanned for this port and attempted to upload code to machines compromised with certain versions of Beagle.

Port 2556 (TCP and UDP) is registered with IANA to nicetec.de (nicetec-nmsvc), as is port 2557 (nicetec-mgmt).

-Extensions Beagle looks for when searching for email addresses

Beagle.A searches for email addresses in files with the following extensions:
WAB, TXT, HTM, HTML.

Beagle.K expands this and searches for email addresses in files with the following extensions:

WAB, TXT, MSG, HTM, XML, DBX, MDX, EML, NCH, MMF, ODS, CFG, ASP, PHP, PL, ADB, TBB, SHT, UIN and CGI.

-DNS servers coded into the worm:*Search results for: 151.201.0.39*

OrgName:	Verizon Internet Services
OrgID:	VRIS
Address:	1880 Campus Commons Dr
City:	Reston
StateProv:	VA
PostalCode:	20191
Country:	US

Search results for: 217.5.97.137

OrgName:	RIPE Network Coordination Centre
OrgID:	RIPE
Address:	Singel 258
Address:	1016 AB
City:	Amsterdam
StateProv:	
PostalCode:	
Country:	NL

-Websites that Beagle sends “GET” information to:

Beagle.A

<http://www.elrasshop.de/1.php>
<http://www.it-msc.de/1.php>
<http://www.getyourfree.net/1.php>
<http://www.dmdesign.de/1.php>
<http://64.176.228.13/1.php>
<http://www.leonzernitsky.com/1.php>
<http://216.98.136.248/1.php>
<http://216.98.134.247/1.php>
<http://www.cdromca.com/1.php>
<http://www.kunst-in-templin.de/1.php>
<http://vipweb.ru/1.php>
<http://antol-co.ru/1.php>
<http://www.bags-dostavka.mags.ru/1.php>
<http://www.5x12.ru/1.php>
<http://bose-audio.net/1.php>
<http://www.stngdata.de/1.php>
<http://wh9.tu-dresden.de/1.php>
<http://www.micronuke.net/1.php>
<http://www.stadthagen.org/1.php>
<http://www.beasty-cars.de/1.php>
<http://www.polohexe.de/1.php>
<http://www.bino88.de/1.php>
<http://www.grefrathpaenz.de/1.php>

<http://www.bhamidy.de/1.php>
<http://www.mystic-vws.de/1.php>
<http://www.auto-hobby-essen.de/1.php>
<http://www.polozicke.de/1.php>
<http://www.twr-music.de/1.php>
<http://www.sc-erbendorf.de/1.php>
<http://www.montania.de/1.php>
<http://www.medi-martin.de/1.php>
<http://vvcgn.de/1.php>
<http://www.ballonfoto.com/1.php>
<http://www.marder-gmbh.de/1.php>
<http://www.dvd-filme.com/1.php>
<http://www.smeangol.com/1.php>

Beagle.B

www.strato.de/1.php
www.strato.de/2.php
www.47df.de/wbboard/1.php
www.intern.games-ring.de/2.php

Beagle.C, .D, .E

<http://permail.uni-muenster.de>
<http://www.songtext.net/de>
<http://www.sportscheck.de>

Beagle.F, .G, .H, .I, .J, .K

<http://postertog.de/scr.php>
<http://www.gfotxt.net/scr.php>
<http://www.maiklibis.de/scr.php>

Beagle.U, .V

<http://www.werde.de/5.php>

Beagle.W

www.lowenbrau.ru
www.ctn.ru
alfinternational.ru
www.psnr.ru
www.deadlygames.de
www.o-problemo.de
www.tv87.de
www.ranknet.de
www.joerrens.de
www.bbszene.de
www.gebr-wachs.de
www.lords-of-havoc.de

comdat.de
www.eurostretch.ru
mir-auto.ru
artesproduction.com
www.hhc-online.de
gaz-service.ru
rdwufa.ru
www.komandor.ru
www.mirage.ru
prizmapr.ru
avistrade.ru
service6.valuehost.ru
www.thomas-we.de
partiyazerna.lgb.ru
pvcps.ru
monomah-city.ru
mir-vesov.ru
promco.ru
www.13tw22rigobert.de
die-cliquee.de

Beagle.X

<http://bohema.amillo.net>
<http://abc517.net>
<http://www.abc986.net>

-Remote Deletion String

Beagle (through version .K) could be remotely removed (files will be deleted, Registry keys remain intact) by sending the following text string to the backdoor port (discovered by Joe Stewart of Lurhq)²⁵:

```
0x43 0xff 0xff 0xff 0x00 0x00 0x00 0x00 0x04 0x31 0x32 0x00
```

-Ties to Mitglieder

The similarities to the Trojan known as Mitglieder are significant. At this time it is impossible to tell if the same author(s) wrote the code, if the common traits were placed in Beagle as a red herring, or if it is all entirely coincidence. Possibly the most telling evidence that the two are linked is Beagle.L and .M utilized a new variant (not seen up to that point) of the Mitglieder code to install a backdoor and turn machines into email relays.

Mitglieder was discovered in the wild January 8, 2004. It has also gone through a number of iterations in its refinement cycle. It opens a mail relay on infected machines. The

Trojan also attempts to download and execute a keystroke logger/password stealer named Ldpinch (sends data to xxx234@mail.ru, compressed in FSG, Visual Basic).

Mitglieder copies itself to the Windows system directory as "ibot4.exe." It adds the value ssgrate.exe to the Registry (a similar entry to the one used by Beagle variants, "ssate.exe" and "srate.exe") and attempts to stop the same process as Beagle with the exception of Avltmain.exe and Outpost.exe. The Trojan attempts to connect to a website and PHP file. The second version of the code turned the infected machine into an email relay.

-Example of Possible Beagle.J/.K Email

Mass mailer worms and spam often employ well-crafted message subjects and text to fool users into opening attachments. Below is an example of a possible email carrying the Beagle.J/.K worm, one of the more successful versions of the code at creating a message that readers would find believable:

From: support@<recipient's_domain>

Subject: Email account utilization warning.

[Message Text]: Your e-mail account will be disabled because of improper using in next three days, if you are still wishing to use it, please, resign your account information.

For details see the attached file.
For security reasons attached file is password protected.
The password is "<random_5-digit __password>".

The <recipient's_domain> team
http://www.<recipient's_domain>

-Icons for Selected Versions

The attachment icons for Beagle variants were undoubtedly part of the social engineering strategy of the author(s). Icons employed for the code include:

Beagle.A



Beagle.B



Beagle.C



Beagle.E



Beagle.G



Beagle.J



Beagle.U



Beagle.V



-Hidden Text Log

A side story to the worm outbreaks of early 2004 is the inclusion of messages, seemingly to other virus writers, as part of the malicious code²⁶. Beagle's part of this dialog is provided in the context of itself, MyDoom, and Netsky:

Beagle.J contains the following line of text, unseen by a user as the worm executes:
"Hey, NetSky, fuck off you bitch, don't ruine our bussiness, wanna start a war ?"

The MyDoom.G variant includes:

"To netsky's creator(s): imho, skynet is a decentralized peer-to-peer neural network. we have seen P2P in Slapper in Sinit only. they may be called skynets, but not your shitty app."

Netsky.F included the following (released the following day):

"Skynet AntiVirus - Bagle - you are a loser!!!!".

Beagle.K "responds" with:

"Hey, NetSky, fuck off you bitch!"

Netsky.G:

"Netsky AntiVirus - Give up, bagle & mydoom, dude! You are fucking your mother! I want to meet you in the U,S,A, Road-App time enc:[fg.od.jgij], and the you will know what pain is"

Netsky.H

"Skynet AntiVirus - MyDoom and Bagle are children"

Netsky.I

"Skynet AntiVirus - MyDoom and Bagle are spammer"

Netsky.J

"be aware! Skynet.cz - -->AntiHacker Crew<--"

Beagle.L

"#####
Hey, NetSky, fuck off you bitch!"

Netsky.K

"Skynet AntiVirus - We want to destroy malware writers business, including MyDoom & Bagle. To F-Secure and so on, we do not want damage systems, we only want to avoid that Bagle continues his dirty business. We have respect of your work (Your heuristic scan is not good enough! Make it better). When the beagle and mydoom loose, we wanna stop our activity. thats now. And personal words to mydoom: Your are so shitty i never seen in my life. A Sample is bin laden and saddam. Your are more, more as more. worse than bad, the only worst. I cannot describe you, you're so lame. And to the mydoom thieves: You will go into the prison next time in texas, nice to meet the bagle author there. Eat my shit, its similar your food, you know. And do not watch too much porn. Last words to all AV firms: We are the Skynet, not netsky! You can use commands on port 26 to deactivate the Skynet!. This is the last version of our antivirus. The source code is available soon. Note that the optimization limit is also reached. You can't get more with smtp engines. bagle and mydoom can continue his dirty impact. the 11th of march is the skynet day."

Beagle.M

“

The White Rabbit Presents

The first and the single
Anti-NetSky AntiVirus



”

Beagle.S

Contains the same picture as above, with the text changed to:

“Yeah, I’m the sneaky thingie ;)”

Netsky.R

"Yes, true, you have understand it.

Bagle is a shitty guy, he opens a backdoor
and he makes a lot of money. Netsky not, Netsky
is Skynet, a good software, Good guys behind it.

Believe me, or not.

We will release thousands of our
Skynet versions, as long as bagle is there and the
people...

Thanks to Bruce Schneider.

And to all people in cz and russia.

Best regards - We are the only SkyNet."

References/Notes

Notes

- 1 The three worms kicked off the 2004 mass mail surge and combined for a higher number of reported viral incidents than all the worms of 2003. Source: <http://www.vnunet.com/News/1153550>
- 2 <http://www.sophos.com/virusinfo/analyses/w32baglea.html>
- 3 Panda Software's Report on Beagle.A
http://www.pandasoftware.com/virus_info/encyclopedia/overview.aspx?IdVirus=43789&sind=0
- 4 A few analysts have submitted the possibility that the worm (and ones like it) are the work of professional spamming outfits. One such report can be found at Sophos' site:
<http://www.sophos.com/virusinfo/articles/wormwarwords.html>
- 5 CA's analysis of MiMail <http://www3.ca.com/virusinfo/virus.aspx?ID=36092>
- 6 There are numerous sites listed below to investigate the Beagle variants described in this report, to see a detailed account of the first version:
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BAGLE.A or
<http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.a@mm.html>
- 7 MyDoom damage prognostication http://www.worldnetdaily.com/news/article.asp?ARTICLE_ID=36800
- 8 Netsky's damage estimate at <http://www.entmag.com/news/article.asp?EditorialsID=6142>
- 9 Mitglieder was detected on some customer machines as reported by Symantec:
<http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.a@mm.html>. This detection may have been testing of the Mitglieder code on infected machines. The reports of finding Mitglieder could have served as a warning to the Beagle author(s), propelling them to stealthier tactics such as removing the code altogether once a test was complete (as is observed as possible in the next version of the code, Beagle.B).
- 10 This possibility is suggested based on the work of Dr. Adam Young and Dr. Moti Yung. The use of cryptography in virus code is exceptionally well documented in their book, "Malicious Cryptography: Exposing Cryptovirology," Wiley Publishing, 2004. ISBN: 0-7645-4975-8.
- 11 The encrypted attachments could not be scanned by many products:
<http://www.techworld.com/news/index.cfm?fuseaction=displaynews&NewsID=1120>.
- 12 Tofger, although it had no self-replicating features, did employ password-protection:
<http://www.enterpriseplanet.com/security/news/article.php/3111701>.
- 13 Versions of Mitglieder discovered after Beagle.K:
<http://www.symantec.com/avcenter/venc/data/Trojan.mitglieder.d.html>
<http://www.symantec.com/avcenter/venc/data/Trojan.mitglieder.e.html>.
- 14 Microsoft's advisories: <http://www.microsoft.com/technet/security/bulletin/MS03-040.asp>.
- 15 Introduction to HTA files:
<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/hta/overview/htaoverview.asp>.
- 16 A good example of the multiple reports needed for the components of Beagle.Q:
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=HTML_BAGLE.Q-1
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=HTML_BAGLE.Q
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=VBS_BAGLE.Q
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=PE_BAGLE.Q
- 17 Beagle.U's look was noted at Trend Micro:
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BAGLE.V&Vsect=T
- 18 Beagle.U took a mere 6 hours to go from Category 2 to 3 according to Symantec:
<http://www.symantec.com/avcenter/venc/data/w32.beagle.u@mm.html>, and reached "epidemic" levels according to Panda Software:
http://www.pandasoftware.com/virus_info/encyclopedia/overview.aspx?IdVirus=45878&sind=0.
- 19 Panda had no posting of Beagle.X/Mitglieder five days after its discovery. Sophos posted a notice that although they had not received a report of the Trojan, they would print a warning based on customer inquiry (as of April 12, 2004 this was: "At the time of writing, Sophos has received no reports from users affected by this Trojan. However, we have issued this advisory following enquiries to our support department from customers." at <http://www.sophos.com/virusinfo/analyses/trojbaglex.html>). The report from F-Secure includes the text of the spammed message: http://www.f-secure.com/v-descs/mitgl_ai.shtml.

20 The similarity to SoBig has been noted by other reports as well: http://news.com.com/2100-7349_3-5143726.html. In addition, the following provide background on SoBig that were used in drawing this comparison: <http://securityresponse.symantec.com/avcenter/venic/data/w32.sobig.a@mm.html>, and the methodology of the worms noted at: <http://www.eweek.com/article2/0,4149,1460087,00.asp>.

21 Jeefo was not especially widespread, but has an interesting infection routine. Details are available in the Symantec report at: <http://securityresponse.symantec.com/avcenter/venic/data/w32.jeefo.html>.

22 SoBig was also believed to be a spammer-created, for-profit worm: <http://news.com.com/2100-1002-5067886.html?tag=nl>

23 Justice can be done to the extensive work done by SEI on the CMMI at:

<http://www.sei.cmu.edu/cmmi/general/general.html>

24 And some considered stopping mail altogether: <http://www.nwfusion.com/news/2004/0202worm.html>.

25 Lurhq can be found at <http://www.lurhq.com>. A good discussion of the remote removal is available at <http://www.f-secure.com/v-descs/bagle.shtml>.

26 The hidden text discussion is expanded at the following locations:

<http://www.sophos.com/virusinfo/articles/wormwarwords.html>,

<http://www.eweek.com/article2/0,1759,1541831,00.asp>, and http://zdnet.com.com/2100-1105_2-5168983.html among others.

Acknowledgements

The details of each Beagle strain were compiled from reports found at the AV vendor sites listed below and independent evaluation of code samples.

Virus identification and reverse engineering produces varied results; exact names will not be consistent with the paper (for internal consistency, the Symantec nomenclature was used for the Beagle variants through .M), however, the following sites will provide a wealth of background on all the variants discussed here.

The Symantec Deep Sight reports were especially helpful with comparing each version of the worm for distinctions. <http://tms.symantec.com>

Symantec Security Response

<http://www.symantec.com/avcenter/>

Trend Micro Virus Information Page

<http://www.trendmicro.com/vinfo/>

Computer Associate's Anti Virus Site

<http://www3.ca.com/virusinfo/>

F-Secure's Virus Information Site

<http://www.f-secure.com/virus-info/>

Panda Software's Virus Information Site

http://www.pandasoftware.com/virus_info/

Sophos

<http://www.sophos.com/>

Kaspersky Labs

<http://www.kaspersky.com/>

Specific Reports of Interest on Beagle

Computer Associate's Report on Beagle.E

<http://www3.ca.com/virusinfo/virus.aspx?ID=38437>

Network Associates' Report on Beagle.H

http://vil.nai.com/vil/content/v_101068.htm

Sophos – Beagle.I Report

<http://www.sophos.com/virusinfo/analyses/w32baglei.html>

Kaspersky Labs' Beagle.A Report

<http://www.avp.ch/avpve/worms/email/bagle.stm>

F-Secure Security Information Center

<http://www.f-secure.com/virus-info/>

F-Secure's Bagle.I Report

http://www.f-secure.com/v-descs/bagle_i.shtml

Panda Software's Virus Information Site

http://www.pandasoftware.com/virus_info/

Trend Micro's Beagle.W Report

http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BAGLE.W

CA's Mitglieder.AC Report (Beagle.X)

<http://www3.ca.com/threatinfo/virusinfo/virus.aspx?id=38807>

Fisher, Dennis. "Viruses tag along," eWeek, March 29, 2004. Volume 21, #13, pg. 25.

Additional Reading

CERT Advisory on E-mail Worms

<http://www.cert.org/advisories/CA-2004-02.html>

The Search for ESA's Beagle 2 Mars Rover

<http://edition.cnn.com/2004/TECH/space/03/08/mars.beagle.reut/>