So You Want to Take Over a Botnet...

David Dittrich
Applied Physics Laboratory
University of Washington

Abstract

Computer criminals regularly construct large distributed attack networks comprised of many thousands of compromised computers around the globe. Once constituted, these attack networks are used to perform computer crimes, creating yet other sets of victims of secondary computer crimes, such as denial of service attacks, spam delivery, theft of personal and financial information for performing fraud, exfiltration of proprietary information for competitive advantage (industrial espionage), etc.

The arms race between criminal actors who create and operate botnets and the computer security industry and research community who are actively trying to take these botnets down is escalating in aggressiveness. As the sophistication level of botnet engineering and operations increases, so does the demand on reverse engineering, understanding weaknesses in design that can be exploited on the defensive (or counter-offensive) side, and the possibility that actions to take down or eradicate the botnet may cause unintended consequences.

1 Introduction

Computer criminals regularly construct large distributed attack networks comprised of up to millions of compromised computers around the globe. This is just the first step and these the primary victims. Once constituted, these attack networks are used for other computer crimes, creating yet other sets of secondary victims of computer crimes such as denial of service attacks, spam delivery, theft of personal and financial information for performing fraud, exfiltration of proprietary information for competitive advantage (industrial espionage), etc.

At one extreme of the spectrum of attacks are large-scale distributed denial of service (DDoS) attacks that are designed to disrupt services. Sites targeted for DDoS attack are typically caught by surprise. The first indication they have that they are under attack are reports of the network being completely unavailable, or the web site itself being non-responsive. It is common for victim sites to be unprepared to collect network or host log information that can assist in determining the full set of attacking IP addresses, or to be able to differentiate attacking hosts from those of legitimate users.

At the other extreme are criminal gangs casting a wide net across hundreds or thousands of sites, slowly and carefully identifying proprietary information, financial data, login credentials, or other types of intellectual property to steal and exploit. A widely reported example is the set of GhostNet intrusions investigated by Canadian and U.S. researchers in 2008 and 2009. In this type of attack, there may be no overt indicators. The attackers *do not want the victims to know* they are victims. Often such attacks last months or years, carefully crafted to fly "under the radar" of the AV industry and victims' own IDS/IPS systems. This is easier to do than many wish you to believe.

1.1 Terminology

The terms *bot* and *botnet* derive from the Internet Relay Chat (IRC) world, which uses a central C&C structure based on human-readable (known as *cleartext*) commands on a single fixed server port. They are widely used today to describe any/all malicious intruder attack networks, despite being essentially content-free, telling you practically nothing about the actual role of each computer involved in a distributed attack network [9] and how to deal with them. Like the term *zombie* they replaced, their primary utility is making it easy for non-technical people to conceptualize a threat. Using these terms in a technical sense of countermeasures and response actions actually causes confusion, hinders understanding *what to do*, and is antithetical to a science. The terms also imply something about topology and C&C method that increasingly leads first responses down useless paths.

Consider the *Storm botnet* that appeared in January 2007. Researchers were interested in Storm's use of a Distributed Hash Table (DHT) based P2P file sharing network to store files indicating the location of C&C servers, from which infected computers *pull* commands. Despite claims of some researchers and non-technical news reporters, Storm did not use P2P to *push* commands (as did Nugache, which appeared a year earlier) for its C&C. Rather, Storm used P2P file sharing for obfuscating its central C&C servers as an alternative (and addition) to the DNS-based *Fast Flux* mechanism also employed for a time. Storm did not use IRC, so its topology is not single *hub and spoke*, but would more accurately be described as multiple hub and spoke (neither

central C&C, nor P2P). At the time the terms *bot* and *botnet* were coined, in almost no sense would Storm be considered either. Anyone monitoring IRC bot channels and running to search their logs would see no commands going to these bots.

While IRC-based attack networks still exist in large numbers, many of the most sophisticated and harmful attack networks share almost no C&C characteristics with IRC-based botnets (i.e., they aren't cleartext, they don't use fixed ports, they are heavily encrypted, the don't rely on other protocols like DNS, and they have no central C&C server). In this paper the term *botnet* is used, begrudgingly, due to its prevalence. The security industry and research community may find it advantageous to adopt more sophisticated terms that carry meaning with them, as was the hope when the terms *distributed system intruder tools* (DSIT), *handler*, and *agent* were carefully chosen by attendees of the first-ever CERT/CC workshop in 1999 [31] to cover a wider range of malicious tools and topologies than just the simple central C&C IRC-based botnet.

1.2 Spectrum of Response Actions

The arms race between criminal actors and the computer security industry/research community who actively try to counter botnets is constantly escalating in aggressiveness. Botnet C&C topologies are getting more complex and hardening mechanisms improving, resulting in increasingly more resilient botnets over time. As the sophistication level of both engineering and operations increases, so does the demand on reverse engineering necessary to produce technical analyses [11, 21, 28, 34] with sufficient detail to acquire a deep enough understanding of weaknesses in design and alternatives to successfully exploit in defensive (or counter-offensive [40]) actions.

Dittrich and Himma describe an *Active Response Continuum* (ARC) in response to computer and network attacks [14]. The ARC describes two continuums, one of *Levels of Response Capacity* and another of *Levels of Force* or aggressiveness of actions. These spectra cover inept to expert response capability, as well as actions ranging from passive observation through retaliatory counter-strikes. The latter spectrum is the primary focus of this work, however the inability (or unwillingness) of entities to respond comes into play as well.

The fundamental problem in dealing with distributed attacks is that of coordinating a response involving data collection, analysis, and countermeasures, across a heterogeneous population that was neither formed as a team, nor under any obligation or requirement to behave as one. The inefficiency of response produces a great temptation for those victims who are in a position of technical knowledge and capability to *do something* unilaterally to stem the attack. The question is whether those actions are both (a) sufficient to achieve the desired objective and (b) more likely to do good rather than harm.

1.2.1 Levels of Response Capacity

In any large-scale, widespread incident involving thousands of systems you will find that some of the sites involved are very responsive and skilled at response, while others are non-responsive, be it due to resource limitations, policy, lack of skilled staff, etc. This creates differentials in capacity to respond that can significantly slow down an investigation, or tempt consideration of taking aggressive unilateral actions of various types. Entities may also be purposefully non-responsive, effectively assisting criminals to remain active.

1.2.2 Levels of Force or Aggressiveness

Along this spectrum there exists a range of specific response actions involving distributed intruder attack tools. At one end is passive observation with little direct interaction. Even this low end option raises communication privacy issues. At the most aggressive extreme of the spectrum is the complete eradication of bot software from all infected hosts. In between are a range of actions involving engagement with the C&C infrastructure and capabilities of infected agent nodes to some degree, all with increasingly thorny legal and ethical questions that are increasingly being raised [1, 26].

Passive Observation: Having visibility into communications between infected agent computers and any C&C infrastructure handling them, in order to identify malicious activity through *IP reputation watchlists* is commonly the first line of defense for network engineers or network operators in service provider or transit networks. Passive monitoring of C&C traffic on botnets is done by many individuals and groups, each with different roles, responsibilities, and authorities. There are potential issues with violating electronic communications privacy laws (e.g., the Wiretap Act in the United States).

Passive observation is fine for distributed intruder networks that employ clear text protocols, especially those using fixed ports or signatures readily detectable by IPS. In some cases, NetFlow alone can give a relatively accurate picture of attack network topology, C&C infrastructure, and potential weak points. Setting aside the potential problems of accidentally violating protected private communications, it may be impossible to determine the totality of compromised hosts by monitoring C&C traffic alone [29]. Only a subset of infected hosts may be active in a single channel or report anything to the channel. The identities of hosts may be obfuscated, preventing direct and unique identification of these hosts.

If the botnet uses heavy encryption, and possibly a P2P infrastructure, passive observation becomes practi-

cally useless. For example, the random topology of the Nugache P2P botnet limited the number of connections to remote peers to no more than a dozen or so per day, requiring a week of observation to see a few hundred nodes in the network, or months of observation to potentially see more of the botnet. Knowing the full size of the botnet from passive observation alone is practically impossible. It was not identifiable reliably by IPS signatures, and nothing useful could be inferred about command activity by looking at the encrypted streams. As distributed attack networks move to use of peer-to-peer mechanisms for C&C, and do a better job of using encryption to conceal the content of communications, passive monitoring also becomes of little use [38, 13]. While used successfully today, this means of countering botnets is becoming less and less viable over time.

Infiltration: Execution of malware in a sandbox environment is common practice. The malware analyst does very little, if anything, other than run a malware sample in a sandbox (e.g., a virtual machine, an emulator, or a bare-metal computer) for a short period of time and look at the results. The malware does its own thing, as if it had infected a real victim's computer. Automated analysis illuminates the internal activity of malware on a host, including modifications to the Windows registry, creation of files, opened network sockets, downloading secondary files, in addition to anything observable on the network by passive observation. Sandbox environments readily can extract the IRC server name, channel name, and login credentials, and a great deal of knowledge about botnet function can be gained by collecting and processing the data [9].

Sandboxes do not fully mimic human behavior, however. Malware run in a automated fashion does not involve human interaction: No keystrokes are logged, no passwords collected, no webpage visits are hijacked. A researcher sandbox may be easily detectable by malware operators because the computer does not exhibit any typical all behaviors that a desktop or home computer would exhibit. In fact, researchers investigating Ozdok were able to find screen captures uploaded to Ozdok's central servers which could allow the botnet operators to select hosts – or avoid, even potentially attack, them – by looking at what applications are visible in screenshots.

Manipulation: Once login credentials have been obtained that allow authentication to the C&C infrastructure, it is possible to move on to manipulation. Manipulation can mean many things, but implies more active engagement than simple infiltration or using captured credentials to initiate sessions with the botnet or related interactive shell accounts. Manipulation involves actively controlling the botnet and causing agents to *do things*, such as causing a dialog box to pop up like BBC reporters did with botnet they leased in 2009 [25]. Actions that occur during manipulation by defenders or researchers are, for all practical purposes, indistinguishable from those of the criminals who established the botnet.

Takeover: The most basic, unsophisticated botnets only employ account/password authentication mechanisms that are quite easy to take over. These botnets are small in size and operated by relatively inexperienced botmasters with rudimentary knowledge. Even some botnets operated by sophisticated adversaries may use simple C&C mechanisms and protections.

Once a sufficiently high degree of certainty exists that one knows the full command set and capabilities of a distributed attack agent, valid credentials allowing one to gain administrative access to the C&C mechanism of a distributed malware network, carefully chosen actions can successfully expose the attack agents without the attacker knowing this is happening. Such detailed knowledge is not easy to come by, and requires a significant investment in reverse engineering and analysis of host and network data from real intrusion events in order reach the level of knowledge required. There is limited number of individuals and groups possessing the skills and data necessary to work at this level of sophistication countering sophisticated attackers. There is also some risk of being detected and counter-attacked, which anyone engaging in this activity should reasonably anticipate and prepare for.

Takedown: Taking down a botnet entails identifying weaknesses in the C&C structure and fall-back rendezvous mechanisms such that you can completely disrupt any new infections, any connections with the C&C infrastructure, and any means of the attacker countering your actions. A series of such takedown operations (some successful and some not) are described in greater detail in Section 2. As one commenter put it, when takedown operations are not successful the botnet operators, "take a break and revise their code to be smarter, faster, and stealthier." [27]

Eradication: Eradication involves not only effecting a takedown, but also using captured C&C capabilities, or remotely exploitable vulnerabilities found in the malware or it's host operating system, to control infected nodes to clean up the malicious software on those nodes on command.

In 2008, researchers from the University of Bonn were among many studying the Storm botnet at the time. Unlike others, the Bonn researchers were able to identify software coding bugs in the Storm bot that allowed them to take full control of infected nodes, identify the running Storm thread and kill it, and download/execute arbitrary code on the infected node. This research was presented at the 25th Chaos Communication Conference (25C3) in Germany in December, 2008 and partial

source code requiring advanced programming skills was released publicly on the `full-disclosure` mailing list. This proof-of-concept demonstrated, at least for a limited population of Storm-infected computers running the version and patch level of Windows with which they developed and tested their code, that it was possible to remotely cleanup some Storm-infected computers without any interaction or intervention of the owners of those systems. This is a risky endeavor that is the subject of some well-reasoned ethical debate [15].

The Conficker Working Group (CWG) report of "Lessons Learned" [32] says,"[the] group has had several outside the box discussions of potential ways to remediate Conficker, but have been hampered by the lack of authority or resources to do so." Eradication (ARC Level 4) raises significant legal and ethical questions, demanding a degree of technical capability, planning, and execution that does not exist today.

2 Takeover/Takedown Case Studies

We will now look at a set of representative botnet takeover and takedown activities that received wide media attention at the time (at least within the computer security community). Each case highlights some of the attributes of the botnet in question, as well as what actions were taken against the botnet and how the botnet operators responded.

• **Torpig** (a.k.a. **Sinowal** and **Anserin**) is a keylogging botnet using a central C&C model with keylog deposition sites obscured by a Domain Generation Algorithm (DGA). It was first reported in February 2006 and received international press coverage for its financial information theft activity in 2008 [33]. It often was accompanied by the **Mebroot** rootkit, with which it is sometimes confused. Estimates of the number of infected hosts range into the millions.

In January of 2009, almost three years after first discovery, researchers at University of California Santa Barbara (UCSB) used information gained from reverse engineering the C&C server selection protocol to identify as-yet unregistered domains that the bots would start using for depositing their keylog files. They registered these domains before the attackers could, set up their own servers in a co-location provider known to be unresponsive to complaints, and temporarily took control of the botnet for a period of approximately 10 days [37]. The attackers noticed the takeover, updated their botnet to resist this weakness in the future, and took back control.

• **Ozdok** (a.k.a. **Mega-D**) is the name given to a low-profile family of malware described in 2008 by Joe Stewart [35]. According to Stewart, Ozdok was so poorly classified by the AV industry that it was known by many generic names that were not recognized as part of a larger coherent pattern of criminal activity. Stewart reported Ozdok was controlled by small set of central C&C servers with no change in IP addresses for over six months.

In 2008, the primary sponsor of Ozdok bot activity was shut down when the Federal Trade Commission (FTC) obtained a court order to freeze the spamming organization's assets and shut down the network [36]. The botnet was simply moved and came back in 2009 to again be one of the top sources of spam.

In November, 2009, FireEye initiated a second takedown operation aimed this time at the primary and fall back domains used by the Ozdok botnet infrastructure. Their operation consisted of (a) notification of involved ISPs, (b) working with registrars to cooperatively take down C&C domains, and (c) registration of as-yet unused domains (similar to what was done by UCSB researchers with Torpig and the Conficker Working Group with Conficker).

• The **Mariposa** botnet was first identified by Defense Intelligence in 2009 [41]. Samples found in the DI report are known by AV engines as (among other names) Malware.Pilleuz [PC Tools], Worm:Win32/Rimecud.A [Microsoft], Packed.Win32.Krap.af [Kaspersky Lab], W32/Autorun.worm.zzq [McAfee], and Mal/Krap-E, Mal/Zbot-I [Sophos]. Operators of the Mariposa botnet concealed their access to central C&C servers behind anonymous VPN services to prevent traceback. It supports some general capabilities including update, download and execute (to install other malware), spreads itself using multiple exploits, and even has a *remove* command [41].

The *Mariposa Working Group* was established to counter the botnet. In December 2009, 7 months after the bot was first observed, the working group began wrestling back and forth with the botnet operators for control of the botnet. The botnet operators retaliated with distributed denial of service (DDoS) counterattacks directing 900 Mbps of attack traffic at the Working Group members, disrupting network services of innocent third parties sharing networks with Working Group members for hours [23]. At one point, one of the leaders of the criminal group made the mistake of attempting to connect to a C&C server without using the anonymous VPN, exposing his personal IP address and identifying him. This information was handed information over to Spanish law enforcement, who subsequently arrested the suspect.

• The **Waledac** botnet was first discovered in April 2008 [39]. Waledac uses a hybrid topology with top-level central C&C servers, a middle tier of servers using a custom peer-to-peer protocol to distributed peer and proxy lists, and a lower tier of *worker* nodes that typically send spam. Estimates of Waledac infections range

from 20,000 up to 390,000, however documents filed in court by Microsoft only cite the 6,600 active spamming nodes per day that could be accurately measured.

In February 2010, Microsoft's *Operation b49*[24] effectively removed control of the botnet from its operators through a combination of the use of civil legal process and technical means. This was the first time in history that a court had granted an *ex parte TRO* forcing a domain registrar to take 277 top-level domains used as C&C entry points for the Waledac bots out of service. Removing these domains, combined with the poisoning of peer lists in the repeater layer of the Waledac botnet, allowed *all infected bots* to be sinkholed. The technical sinkholing followed methods described in published analyses of Waledac [34, 5]. Microsoft returned to court several months later and on October 27, 2010, was awarded permanent ownership of the 277 domains under a default judgement.

• **Bredolab** (possibly a.k.a., **Harnig**) was first reported in mid-2009 [18]. Bredolab is not a bot, in the classic sense, but more of a framework for dropping many other malware binaries, including Zbot (a.k.a., Zeus), SpyEye, TDSS, HareBot, Blakken (a.k.a., Black Energy 2), and others. It uses a Fast Flux technique to spread connections from infected computers across a large number of non-linked C&C servers, over which commands are requested (pulled) by bots using HTTP requests [18]. Infected nodes are managed using a *control panel* similar to malware like Zeus [7]. The Dutch Police estimated the number of infected nodes to have been 30,000,000.

On October 25, 2010, the Dutch High Tech Crime Team, part of the National Crime Squad, announced they had shut down 143 Bredolab botnet control servers and taken control of the botnet and had identified and assisted in the arrest of the person suspected of operating the Bredolab infrastructure [20]. They used this ability to push a very simple program (designed to have the least possible risk of harm) to infected computers requesting software to download. This program was the technical equivalent of the *Hello World!* program, doing nothing other than popping up a window explaining the computer was infected and providing an URL to a web page where instructions on how to clean the infection were found. As many as 100,000 users followed the link to display the web page, with as few as 55 individuals registering formal complaints about the Dutch Police' actions.

At least one report suggested the Bredolab infrastructure was continuing to spread malware indicate the takedown may not have been complete [7]. Anecdotal evidence suggests Harnig may be a follow-on to Bredolab that appeared within months of the Dutch police actions. Combining that with reports from AV companies conflating the two (see Rustock section and [16]) suggest that Bredolab may have in fact survived the Dutch police actions and at least shares some infrastructure with, if not evolving into, Harnig.

• The **Pushdo / Cutwail** botnet (a.k.a., **Pandex**) was first reported in January 2007. Pushdo/Cutwail has been called one of the most prolific spamming botnets since 2009. MessageLabs estimated the size of the Pushdo botnet to be between 1.5–2 million infected nodes. It combines an advanced dropper (the Pushdo component) and modules, one of which is the Cutwail spam module. The Cutwail component uses a simple custom encryption block-based encryption algorithm with a fixed hard-coded key. Pushdo encrypts the binary modules it downloads with a fixed key that is sent in the HTTP GET request sent to C&C servers. Pushdo itself does not self-propagate. Rather, it has been seen to be dropped by PE_VIRUT, TROJ_EXCHANGER and TROJ_BREDOLAB, in concert with other dropped malware including Storm, Srizbi, Rustock, and AntispywareXP2009.

In August 2010, the startup company *Last Line of Defense* contacted ISPs in control of two-thirds of the Command and Control (C&C) servers used by Pushdo/Cutwail [22]. Knowing they were reliant on cooperative action from possibly unwilling ISPs, they were open about saying they did not expect a full takedown. Their action did reduced spam volumes for approximately 48 hours, but as anticipated, backup C&C servers kicked in and the botnet regained its full strength.

• The **Rustock** botnet was first reported in early 2006 [3]. The B variant (also known as **Spam-Mailbot.c**) was reported in July 2006. It was reverse engineered by multiple parties who published analyses about six months later in early 2007 [4, 8]. Rustock is designed exclusively to send spam and does not use the standard C&C mechanisms of generalized command distribution as other bots. Its central C&C servers and proxies for those controlling distribution of spam were housed in *bullet-proof hosting providers* who would not act on complaints.

On March 16, 2011, nearly five years after initial reports, Microsoft publicly announced that a collaborative effort led by Microsoft's Digital Crimes Unit had successfully taken down the Rustock botnet, again using civil legal process (an ex-parte TRO) and U.S. Marshals to execute search warrants and seizure of evidence for further legal action (*Operation b107* [3]). One security researcher believes that the Harnig botnet – also known as Bredolab by some AV companies and known to be used to distribute Rustock malware – also ceased to be active at the time of the servers being seized with the assistance of U.S. Marshalls[30].

• The **Coreflood** botnet was first discovered in

2001 [43]. This botnet successfully remained under the radar of the computer security industry for years because of its low profile and non-aggressive tactics. Computer security researchers were able to gain cooperation of hosting providers in the United States to get a copy of one of the C&C servers, allowing them to analyze the server and learn about how it functions.

In April, 2011, a U.S. Federal Court granted United State Department of Justice an *ex parte* temporary restraining order (TRO) and orders from the court to allow Internet Software Consortium (ISC) to sinkhole Coreflood bots and take control of them [42]. ISC was granted authority to sinkhole bots and collect information allowing them to identify the owners of infected hosts to whom the FBI could send a *Notice of Infected Computer* and form *Authorization to Delete Coreflood from Infected Computer(s)* that grants the FBI the right to issue "remove" commands to the infected computers. The court explicitly granted the FBI the authority to execute the "stop" command, but not to execute any other commands (including the "remove" command) without express permission from the owners of infected computers via the authorization form.

• The **Kelihos** (a.k.a., **Hlux** and **Darlev**) appeared in December 2010 [2] and is believed to be a re-write of Waledac, due to close similarities in its C&C topology, command structure, and other architectural features. At its peak, Kelihos was believed to have infected 41,000 computers worldwide [6]. Reverse engineers at Kaspersky Labs created decryption programs and fake bots, allowing them to observe how Hlux (as it is known by Kaspersky) functioned and to develop a sinkhole mechanism.

If this is Waledac 2.0, it's appearance occurred ten months after Microsoft's *Operation b49* rendered the previous Waledac botnet inoperative. Just nine months after its appearance, on September 26, Microsoft again obtained a court order to take out the domain names used by Kelihos (*Operation b79* [6]) which allowed the successful sinkholing of all infected bots by Kaspersky Labs [44].

2.1 Observations

The case studies above are summarized in Table 1. Several observations can be made about these takedowns in terms of botnet analysis methods and botnet takedown actions.

2.2 Observations about analysis methods

• Size estimates vary wildly from source to source, sometimes differing by 2–3 orders of magnitude. Most published estimates are not accompanied by counting methodology or time period over which counts were made. There is a huge incentive to inflate botnet size to publicly claim defeating "the world's biggest botnet." Such hyperbole serves interests of self-promotion more than it contributes to countering the botnet threat. Cases like Mariposa, going from no known infections prior to 2009 to 12 million less than a year later, suggest visibility and enumeration methods are fundamentally flawed and that sensational claims deserve substantiation.

• Names matter, as do classifications, when trying to understand what kind of system one is dealing with. The multiple names, unstructured analytic reports, and little longitudinal situational awareness hinders takedown actions.

• Information sharing today is ad-hoc, unstructured, and cumbersome. Data interchange formats like Mitre's MAEC [19] – used by Cuckoo Sandbox [17] and Thug [12] – facilitate automated reporting and analysis, allowing more sophisticated responses.

2.3 Observations about botnet takedown actions

• All takedowns coordinating civil and/or criminal legal process with technical methods (Waledac, Rustock, Coreflood, and Kelihos) succeeded on first try, while those only using civil legal process (the first attempt at taking down Ozdok) or using only technical means (Torpig, Ozdok, and Pushdo) did not. Of the most sophisticated botnets, Waledac, Rustock, Coreflood, and Bredolab could not have been fully taken down without using legal process to remove *all* of the top-level domains used for fall-back and secondary C&C.

• LastLine's efforts against Pushdo support the idea that compelled action via court order may be a key requirement for success. Botnets of similar or lesser sophistication (Torpig, Mariposa, and Pushdo) were not successfully taken over permanently on first try. This could be due to insufficient knowledge of the botnet, insufficient planning, or (more likely) because not all attacker-controlled assets were definitively taken away.

• The majority of the takedowns were initiated years after the malware was first recognized in the wild. This allowed sufficient time for multiple groups, in most cases, to analyze and discuss the botnet in private venues. It may also be the result of the confusion and poor classification seen in many cases.

• The exceptions to the multi-year rule were Kelihos and Mariposa. While the Kelihos takedown involved a significant investment of resources by Microsoft in analysis and legal preparation, a process was already established from the Waledac takedown.

• The Mariposa takedown failed initially and also harmed innocent third parties. It eventually succeeded by chance. It is not clear why the initial attempt failed, nor is it clear why Mariposa, if it truly was the "largest botnet in history [41]," would come out of nowhere in

Botnet	Peak Size (est)	First Seen	Take Down	Time Elapsed	Success on 1st try	Used Legal Process
Torpig	180,000	Feb 2006	Jan 2009	3 years	No	No
Ozdok	264,784 [1]	Early 2008	Nov 2009	2 years	No	No
Mariposa	12 million [2]	May 2009	Dec 2009	7 months	No	No [3]
Waledac	6,600+ [4]	Apr 2008	Feb 2010	3 years	Yes	Yes
Pushdo	1.5-2 Million	Jan 2007	Aug 2010	3.5 years	No	No
Bredolab	30 million [5]	Mid-2009	Oct 2010	1.5 years	No	Yes [6]
Coreflood	378,758 [7]	2001	Apr 2011	10 years	Yes	Yes
Rustock	1.6 million [8]	2006	Mar 2011	5 years	Yes	Yes
Kelihos	41,000	Dec 2010	Sep 2011	8 months	Yes	Yes

Table 1: Botnets subject to highly publicized takedown efforts (by takedown date)

[1] *Unique IPs connecting to FireEye's sinkhole in 24 hrs. The 2008 estimate of 35,000 by Marshal Software [35, 36] provided no time frame or counting methodology.*

[2] *Unique IP addresses over an unspecified time period [10]. Other estimates show no more than 1.5M per day.*

[3] *The Mariposa Working Group did not use legal process in their botnet takedown attempts, but information they obtained was provided to law enforcement who eventually made arrests.*

[4] *Count of actively spamming nodes in 24 hr period.*

[5] *Count of total infections, not to be considered a single monolithic botnet of 30M computers. Also, counting method and time period used to establish count was not specified.*

[6] *Criminal procedures were used to seize control of C&C servers.*

[7] *Unique IP addresses seen over a six month period.*

[8] *Size estimated by Microsoft immediately after court-ordered takedown.*

2009. Because it is a general downloader at heart, it may just be an iteration of a botnet that the security industry and researchers did not recognize and may actually be older.

• The Mariposa Working Group aggressively engaged with the botmasters who regained control and retaliated, harming innocent third parties as a result. Had the attacker not made a mistake, arrests may not have been possible. Since the Mariposa Working Group operated in secrecy, only news reports support public understanding of the legal and ethical reasoning process employed in deciding on the actions to take. There is no apparent outside ethical or legal evaluation or approval of the techniques used by the Working Group, as occurred in those cases involving civil legal process. This issue of opacity (or lack?) of ethical analysis is common.

3 Conclusions

The nature of an *arms race* is that every effective defensive action can result in increased sophistication on the attacker side. On defense, there are (a) commercial entities with narrow profit motives and inadequate malware classification, (b) academic researchers constrained by academic schedules, tight competitive funding, and human subjects protection requirements viewed by some researchers as impediments to their progress, and (c) independent researchers volunteering their time and acting based on their own internal moral compasses. It is easier and cheaper for attackers to invest resources to maintain attack infrastructures than it is, in aggregate, for a large number of uncoordinated and competitive entities to do expensive reverse engineering necessary for effective countermeasures. Over time, the effect is that fewer and fewer entities (by themselves) have the resources necessary to definitively take down the most sophisticated criminal botnets. The application of significant resources by a single entity does not scale and is not sustainable.

As more botnet takedowns are publicized, the temptation increases for individuals with reverse engineering skills to do attempt to "get into the game." Corporations and university research labs may have reputational risks limiting how aggressive they get. Individuals with strongly held beliefs, narrow views of what benefits society, or limited resources, may act based on partial information, or with inadequate planning and testing, putting innocent third parties at increased risk.

The security industry and researchers can, however, step up their efforts to better integrate analyses and coordinate collective action to achieve the required technical objectives. The community must learn to work collaboratively and with greater efficiency in sharing knowledge in order to gain necessary agility.

An even more fundamental underlying issue is the immaturity of our collective understanding of how to balance ethical, legal, technical, and political considerations so as to achieve the collective end-goal of eliminating computer crime threats while simultaneously putting the best interest of society as a whole above the issues just listed. Funding agencies, such as the Department of Homeland Security, are developing ethical guidelines [1]. Security researchers need to engage with these

efforts to ensure that as actions move up the spectrum of aggressiveness, risk of harm to the public that is purportedly being served by our actions does not similarly rise.

References

[1] "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research" ("Menlo Report") for the Department of Homeland Security (DHS), Science and Technology, Cyber Security Division (CSD), December 2011. Docket No. DHS–2011–0074.

[2] ADAIR, S. New Fast Flux Botnet for the Holidays: Could it be Storm Worm 3.0/Waledac 2.0?, December 2010.

[3] ANSELMI, D., BOSCOVICH, R., CAMPANA, T., DOERR, S., LAURICELLA, M., PETROVSKY, O., SAADE, T., AND STEWART, H. Battling the Rustock Threat, June 2011. Microsoft Security Intelligence Report Special Edition.

[4] BOLDEWIN, F. A Journey to the Center of the Rustock.B Rootkit, January 2007.

[5] BORUP, L. T. Peer-to-peer botnets: A case study on Waledac. Master's thesis, Technical University of Denmark, 2009.

[6] BOSCOVICH, R. D. Microsoft Neutralizes Kelihos Botnet, Names Defendant in Case, September 2011.

[7] CHECHIK, D. Bredolab Trojan – Malware Review, December 2010.

[8] CHIANG, K., AND LLOYD, L. A case study of the rustock rootkit and spam bot. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets* (Berkeley, CA, USA, 2007), USENIX Association, pp. 10–10.

[9] CHO, C. Y., CABALLERO, J., GRIER, C., PAXSON, V., AND SONG, D. Insights from the Inside: A View of Botnet Management from Infiltration. In *LEET'10: Third USENIX Workshop on Large-Scale Exploits and Emergent Threats* (April 2010).

[10] CORRONS, L. Mariposa botnet, March 2010.

[11] DECKER, A., SANCHO, D., KHAROUNI, L., GONCHAROV, M., AND MCARDLE, R. A study of the Pushdo / Cutwail Botnet, May 2009.

[12] DELLAERA, A. Low-interaction honeyclient Thug released! https://www.honeynet.org/node/827.

[13] DITTRICH, D., AND DIETRICH, S. Command and control structures in malware: From Handler/Agent to P2P. In *USENIX ;login: vol. 32, no. 6* (December 2007).

[14] DITTRICH, D., AND HIMMA, K. E. Active Response to Computer Intrusions. Chapter 182 in Vol. III, Handbook of Information Security, 2005. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=790585.

[15] DITTRICH, D., LEDER, F., AND WERNER, T. A Case Study in Ethical Decision Making Regarding Remote Mitigation of Botnets. In *Proceedings of the 14th International Conference on Financial Cryptograpy and Data Security* (2010), FC'10, Springer-Verlag, pp. 216–230.

[16] GRIER, C. Anti-virus labels are not suitable for system evaluation. http://www.imchris.org/wp/2011/09/02/, September 2011.

[17] JEKIL. MAEC second round: support completed. http://blog.cuckoobox.org/2012/01/31/maec-second-round-support-completed/.

[18] KADIEV, A. End of the Line for the Bredolab Botnet?, December 2010.

[19] KIRILLOV, I. A., CHASE, M. P., BECK, D. A., AND MARTIN, R. A. The Concepts of the Malware Attribute Enumeration and Characterization (MAEC) Effort. http://maec.mitre.org/about/docs/The_MAEC_Concept.pdf, 2010.

[20] KIRK, J. Dutch team up with Armenia for Bredolab botnet take down, October 2010.

[21] LEDER, F., AND WERNER, T. Know Your Enemy: Containing Conficker, April 2009.

[22] LEMOS, R. What it takes to shut down a botnet. http://www.infoworld.com/t/anti-spam/what-it-takes-shut-down-botnet-903, August 2010.

[23] MCMILLAN, R. Spanish Police Take Down Massive Mariposa Botnet, March 2010.

[24] MICROSOFT NEWS CENTER. Cracking Down on Botnets, March 2010.

[25] MILLS, E. BBC buys, uses botnet to show dangers to PCs, March 2009. http://news.cnet.com/8301-1009_3-10195550-83.html.

[26] MOORE, T., AND CLAYTON, R. Ethical Dilemmas in Takedown Research. In *Financial Cryptography and Data Security*, vol. LNCS 7126 of *Lecture Notes in Computer Science*. Springer-Verlag, 2012.

[27] MUSHTAQ, A. Harnig is Back, August 2011.

[28] PORRAS, P., SAIDI, H., AND YEGNESWARAN, V. Conficker C P2P Protocol and Implementation, September 2009.

[29] RAJAB, M. A., ZARFOSS, J., MONROSE, F., AND TERZIS, A. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets* (2007), USENIX Association, pp. 5–5.

[30] RASHID, F. Y. Harnig Malware Botnet Also Shut Down After Rustock Raid, April 2011.

[31] SEVERAL. Results of the Distributed-Systems Intruder Tools Workshop. CERT/CC, December 1999.

[32] SEVERAL. Conficker Working Group: Lessons Learned, June 2010.

[33] SHIELS, M. Trojan virus steals banking info, October 2008.

[34] SINCLAIR, G., NUNNERY, C., AND KANG, B. B. The Waledac Protocol: The How and Why. In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE) 2009* (February 2010), pp. 69–77.

[35] STEWART, J. Ozdok/Mega-D Trojan Analysis. http://www.secureworks.com/research/threats/ozdok/, February 2008.

[36] STONE, B. Authorities Shut Down Major Spam Ring, October 2008.

[37] STONE-GROSS, B., COVA, M., CAVALLARO, L., GILBERT, B., SZYDLOWSKI, M., KEMMERER, R., KRUEGEL, C., AND VIGNA, G. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of the ACM CCS* (November 2009).

[38] STOVER, S., DITTRICH, D., HERNANDEZ, J., AND DIETRICH, S. Analysis of the Storm and Nugache Trojans: P2P is here. In *USENIX ;login: vol. 32, no. 6* (December 2007).

[39] TENEBRO, G. Threat AnalysisW32.Waledac, 2009.

[40] THE YOMIURI SHIMBUN. Govt working on defensive cyberweapon / Virus can trace, disable sources of cyberattacks. http://www.yomiuri.co.jp/dy/national/T120102002799.htm, January 2012.

[41] THOMPSON, M. Mariposa Botnet Analysis. http://www.defintel.com/docs/Mariposa_Analysis.pdf, October 2009.

[42] UNITED STATES DEPARTMENT OF JUSTICE. United States v. John Does 1 – 13, April 2011.

[43] VAMOSI, R. Security expert talks Russian gangs, botnets, November 2008.

[44] WERNER, T. Botnet Shutdown Success Story: How Kaspersky Lab Disabled the Hlux/Kelihos Botnet, September 2011.