2011 3rd International Conference on Cyber Conflict
C. Czosseck, E. Tyugu, T. Wingfield (Eds.)
Tallinn, Estonia, 2011 © CCD COE Publications

On the Arms Race Around Botnets – Setting Up and Taking Down Botnets

Christian Czosseck
Cooperative Cyber Defence Centre of Excellence
Tallinn, Estonia
christian.czosseck@ccdcoe.org

Gabriel Klein
Fraunhofer FKIE
Wachtberg, Germany
gabriel.klein@fkie.fraunhofer.de

Felix Leder
Institute of Computer Science 4
University of Bonn
Germany
leder@cs.uni-bonn.de

Abstract—**Botnets are a well-recognized and persistent threat to all users of the Internet. Since the first specimens were seen two decades ago, botnets have developed form a subject of curiosity to highly sophisticated instruments for illegally earning money. In parallel, an underground economy has developed which creates hundreds of millions of euros per year in revenue with spamming, information theft, blackmailing or scare-ware. Botnets have become a high-value investment for their operators that need to be protected from law enforcement agencies or the anti-botnet community. Security researchers and companies trying to keep them within bounds are facing the very latest in spreading and defense techniques. Hundreds of thousands of new malware samples per month pose an immense challenge for AV companies. Specialized countermeasures against botnets have evolved along with botnet technology, trying to bring them down by targeting the root of every botnet: its command-and-control structure. This leads to an ongoing arms race between botnet developers and their operators vs. security experts. So far the former have the upper hand.**

Based on the analysis of multiple botnet takedowns and the in-depth investigation of various botnet architectures conducted by the authors, this paper provides an analysis of the efforts needed to acquire and set up a botnet. This is followed by a comparison of selected significant botnet countermeasures, which are discussed with regard to their required resources. Legal and ethical issues are also addressed, while a more thorough discussion of these will be left for future work.

Keywords-IT security; botnet; malware; infection; disinfection; botnet setup; botnet takedown; tactical takedown;

I. INTRODUCTION: CURRENT STATE OF THE ARMS RACE

Botnets are networks of computers infected with malicious software (malware), remotely controlled by so-called bot herders. The infected machines within this botnet (a.k.a. bot or zombie) are regularly abused to perform mostly criminal activities without the knowledge of their owners. This includes but is not limited to sending spam, conducting distributed denial-of-service (DDoS) attacks or harvesting sensitive data such as credit card credentials. Beyond credit card fraud, extortion schemes can also be observed with threats of large-scale DDoS attacks unless payments are made. All this leads to steadily increasing financial damage and cyber crime's yearly income surpassing the global drugs revenue [1]. As a latest trend, botnets play an increasing role in politically motivated attacks against public and private institutions, sometimes threatening entire countries [2]. Behind this is a well-developed underground market, on which botnet technology and associated services are sold to everyone at rather low prices [15].

Anti-virus (AV) companies as the natural enemy of malware are constantly trying to keep up with the growing threat, developing a variety of products to protect computers from being infected. Unfortunately, malware authors are often one step ahead because of the reactive defense provided by AV software. If newly developed malware is released, even up-to-date anti-virus detectors are often not likely to detect it [3]. Some AV software detects less than 10 % of new samples within the first 24 hours of their occurrence. Often manual analysis of new malware samples is required because automatic approaches are limited in their capabilities. There is a general consensus in the AV industry that current solutions are neither scalable nor sustainable enough.

Acknowledging the fact that malware spreading cannot be stopped or slowed down significantly, other countermeasures directly attacking established botnets have been developed.

To receive or pass on commands, the individual parts of these botnets need to communicate with each other and with their so-called command-and-control (C&C) servers. The method according to which this communication takes place defines the topology of the botnet. So far three different ones have been established: centralized topologies with few C&C servers, decentralized topologies based on peer-to-peer (P2P) protocols, and semi-flexible topologies often realized by fluxy domain registrations.

If connectivity between bots and C&C servers is established, different communication protocols like HTTP or IRC are commonly used. A more in-depth discussion of technical botnet issues can be found in [4] and [5]. All these technical aspects provide entry vectors for targeted counter measures against botnets.

This paper provides a comprehensive overview of the resources needed in this arms race between bot herders and botnet hunters. Based on analyses of recent botnet investigations and experience from conducted takedowns, the most common countermeasures are presented and analyzed.

The paper discusses the countermeasures with regard to their required resources, namely *required skills*, *monetary costs* and *time* as well as the *likelihood of a* *successful* takedown. Legal and ethical aspects are also addressed. The discussion is based on a simple taxonomy of botnets presented in Section 2, grouping botnets into three broad groups. In Section 3, we discuss the efforts needed to set up a botnet for each of the introduced categories. This is followed by an in-depth discussion on required resources for the most common botnet countermeasures in Section 4. We conclude with a summary and an outlook on future developments.

II. BOTNET EVOLUTION

Since the world fist encountered a computer virus called Brain back in 1984, malware has developed from a proof of creativity to a highly sophisticated instrument usable for various tasks, nowadays aiming mainly for earning money in a criminal way.

Botnets themselves evolved from the idea of a massive remote control for administrative tasks to a flexible type of malware encompassing the most successful spreading and hiding techniques.

Botnets evolved and became more professional over time. This is reflected in their capabilities, but also in the skills needed for their creation. Nowadays, experienced and knowledgeable malware developers are needed to create botnets which are hard to detect or to mitigate.

This paper introduces three broad categories of botnets reflecting the major evolutional steps over the past decades. The discussion on setting up or taking over/down botnets is structured according to these categories.

A. *Open-Source Botnets*

The first category is formed by botnets that were either developed open-source, were made freely available later on, or are easy to find. This marks the very beginning of malicious botnet development, where botnets and malware in general was often written for (often still illegal) fun or out of competition between “geeks”. Monetary aspects were hardly a driving force.

Well-known representatives are botnets like AgoBot, SDBot or RBot [22, 23]. While quite old, they are still in use and are sometimes developed further by single groups adding new exploits or functionality. These new exploits are developed individually or obtained from other sources like the exploit framework Metasploit [6]. For this category we assume that most of the code base is freely available for both malware developers and AV companies. They are easy to set up, typically only requiring the botherder to make some changes in provided configuration files and compiling the code.

An alternative way of operation is the development of a closed-source botnet (which might be a fork of or inspired by an existing open-source botnet), adding well developed open-source components to the code base. Popular examples for open-source components integrated into closed-source botnets are cryptographic and compression routines. Waledac used the OpenSSL library [4, 7], for both RSA and AES, Conficker included the official MIT implementation of the MD6 hash algorithm [8], and Storm made use of the zlib [9] compression library [24]. This provides malware developers with reliable, well tested standard routines, reducing their efforts. If a particular botnet is a fork of one of the older open-source botnets mentioned above, we consider it to fall under this category. Otherwise the botnets belong to one of the following categories.

B. *Construction Kit-based Botnets*

Over time, botnets developed into an effective tool to illicitly earn money. With AV and other security companies putting more efforts into fighting botnets, botnet developers improved resistance to countermeasures. However, they invented an increasing number of new features for generating money, e. g. by harvesting financial credentials or other valuable information and subsequently selling them later. Botnet developers started to understand the value of their creations and that not everybody is able to develop sophisticated botnets, thus raising the value of well-developed ones. Out of this a business model developed in an underground scene, where botnet developers started to sell botnet software. To an increasing extent, this is offered together with patch services, infection guarantees and/or hosting services. Botnets became available as so-called constructions kits, enabling everyone to configure and create their own botnet in a point-and-click fashion.

This category covers all botnets fitting this description. They are assumed to be well maintained, regularly updated, and coming with the ability to add new functionality (add-ons), maybe even by third parties. Many of them are sold including support for the buyer via ICQ or other digital media. These botnets are normally developed as closed source, using state-of-the-art methods, software development processes and quality assurance methods [10]. To protect the botnet software, licensing schemes and code protection software such as VMProtect [11], commonly encountered in legitimate software products, are used to control the distribution of their products. This makes analyzing or stealing the botnet's source code hard for competitors and AV companies.

Prominent examples of botnets that are sold as construction kits are ZeuS and its presumable successor, SpyEye, both targeting financial data. In the case of ZeuS, the C&C server is provided based on open-source components written in PHP. Prices usually range from a few hundred to several thousand USD depending on the feature set [15].

C. *Specialized Botnets*

The last category this paper introduces covers all botnets, which were developed with a very specific target or functionality in mind. While most of the attributes of the second category still apply, specialized botnets are highly professionally developed, combining advanced expertise in exploit development (e. g. by usage of 0-day exploits), careful software engineering considering latest countermeasures, and sometimes even combine cross-domain knowledge or intelligence of the target. Monetary gains as a driving force might but do not need to be present. Espionage and sabotage are other motivations for this advanced persistent threat (APT).

Examples for this category are Ghostnet [12], which aimed at political espionage in China-critical communities, or Stuxnet, which was developed to target Windows-based supervisory control and data acquisition (SCADA) systems and is assumed to be an instrument of information gathering and sabotage of the Iranian nuclear program [13]. Night dragon is a botnet spying mainly on petrol and gas companies [25]. Another example is Conficker, which, while actively developed to be impervious to latest countermeasures and widely spread, still has not shown any active functional payload.

III. SETTING UP A BOTNET

In 2008, spammers alone earned an estimated 780 million USD [14] and there is an upward trend to these numbers. With this ever increasing amount of money to be made by operating or renting out botnets, an increasing professionalization in the domain can be observed [26]. Structures similar to free market (sub-) economies are emerging where prices and the availability of products and services are regulated by demand. There are even marketing campaigns on underground forums promoting certain products. Taking these issues into account, what are the resources that remain to be expended for setting up and deploying a botnet? In this section we will discuss these resources in the context of the botnet categories introduced in the previous section.

A. *Finances*

As the development of open-source botnets is community-driven, no direct monetary cost is involved. Depending on the situation, software developers might need to be paid.

The prices of construction kit botnets vary; entry-level ZeuS kits can be purchased for 3,000-4,000 USD, whereas more advanced kits can cost more than 10,000 USD [15]. Additionally, appropriate infection kits can be bought from 100 to more than 1,000 USD [16]. A range of companies exist that provide “all-inclusive” packages for botnet construction, propagation with exploit kits, as well as command-and-control server hosting and maintenance.

Where specialized botnets are concerned, especially skilled and trusted developers are required. Components are typically self-developed and seldom purchased. The required trust and skill level makes these types of botnets more expensive than extensions to open-source botnets. In case of sabotage and for reliable development, test environments have to be bought, set up, and maintained [13].

B. *Development Skills*

There are three aspects to be considered when developing a botnet: the infection of the target machine, the botnet binary that is executed on the target machine after infection, and the C&C infrastructure. Different development skills are required for each aspect. Resource requirements for infection are fairly similar for the different classes of botnets and are discussed in a subsequent section.

To configure and install the bot component of an open-source botnet, the user needs a basic understanding of source code configuration and needs to be able to use a compiler. A non-technical user can acquire these skills in a short amount of time. However, a risk in this case is the unknown programming quality of the malware.

Compared to this, the configuration and setup of construction kit botnets is almost negligible. These kits are for sale and designed with user-friendliness in mind. The entire process takes only a few clicks of the mouse. Configuration is accomplished easily by adapting existing configuration files or purchasing ready-made ones. Most kits come with a standard set of system manipulations.

With specialized botnets, the greatest difficulty lies in the amalgamation of cross-domain knowledge, . This does not usually apply to botnets in the other groups. Specialized botnets have highly customized goals, e. g. espionage or sabotage. Exploiting weaknesses and optimizing malware for execution in systems in these environments requires a high degree of immersion in that context. An example of this is Stuxnet. Here, a detailed familiarity with very specific industrial control systems was required. The actual technical skills are comparable to those required for open-source botnets, although in most cases there is no software base to build upon so extensive development effort is needed. An additional difficulty is that community support cannot be relied on here.

Where the development of C&C infrastructure is concerned, construction kit-based botnets require the least effort of the three classes. In principle, setting up a C&C server is identical to setting up any other content-management system. For a more in-depth discussion of C&C infrastructures, please refer to [4]. Protecting the C&C server against takedown attempts by authorities and security researchers is more challenging, but often all-inclusive bundles are offered that include setup, support and bulletproof hosting of the C&C server. For open-source and specialized botnets, these activities have to be performed manually.

C. *Defensive Skills*

To protect their software from reverse engineering and analysis, malware authors increasingly employ defensive measures on a technical level.

An often-used mechanism is encryption, both of the communication with the C&C server and of the malware binary itself. Circumvention of the former is always possible. This is an imminent weakness in botnets using encrypted communication because the encryption keys either need to be included in the binary or can be observed when processed in the binary during runtime.

Obfuscation is a technique for hiding that different malware samples belong to the same botnet and to complicate detailed analysis of the internals. A recent trend is so-called server-side polymorphism. Here, the server from which a newly infected machine retrieves the actual malware encodes the binary differently for every client. This can include differences in the encryption routine, encryption keys, etc. The result is that binaries from two different infected hosts have nothing in common at first glance.

Already existing malware can be precisely immunized against certain AV products or analysis tools. This can easily be done manually because of Web sites such as VirusTotal, a meta-anti-virus tool that allows the online scanning of malware samples with multiple AV solutions. Malware can also be hardened automatically using third-party tools.

By implementing blacklists of IP addresses of known honeypot or other analysis systems, malware developers can explicitly avoid infecting malware analysis systems. Knowledge about such systems can be gathered in a variety of ways. A Web site called AV Tracker [18] contains a comprehensive list of sandboxes, Honeypots and other analysis systems operated by the AV industry and malware researchers world-wide. Going one step further, ZeuS operators have been observed to set up a honeypot-like system to analyze and provide further information about researchers trying to infiltrate its administrative interface [19].

In the case of open-source botnets, the employed defensive measures are hardly sophisticated and are mostly self-developed. Sometimes, adaptations of standard mechanisms can be observed. Botnets made with construction kits typically either have the defensive measures integrated into the construction kit or make use of so-called defensive kits. This modular technique allows the integration of arbitrary defensive measures into the construction kit just before the malware binary is compiled. Depending on the sophistication of these defensive kits, they are either freely available or need to be purchased. Defensive measures for specialized botnets are normally a mixture of standard techniques along with custom-built developments that ensure the stealth properties of the malware binary.

Because security researchers actively study and circumvent these defensive mechanisms, the result is a constant arms race in which botnet operators and developers continually develop new and more advanced mechanisms which are then analyzed and bypassed by the security industry.

D. *Deployment*

Originally, malware spread by exploiting server services via portable disks, nowadays often USB sticks.

A new trend is the increasing exploitation of vulnerabilities in client-side applications. These are often ubiquitous on user desktops and thus an easy target. Examples for these kinds of applications are Adobe’s Portable Document Format (PDF) reader and Flash, Microsoft Office programs, or Web browsers. The latter can be exploited by so-called drive-by downloads on infected Web sites. These Web sites can be both legitimate sites hacked by criminals, or sites especially set up for the express purpose of infection. In the latter case, mass spam e-mails containing links to these sites lured users to these sites. According to the Websense 2010 threat report, 79.9 % of Web sites with malicious code were compromised legitimate sites [20].

Lately, there has been an increase in so-called targeted attacks which contain a social engineering component. Detailed background information is gathered on the intended targets and personalized messages are sent to the victims, either via email or through social networking sites. By exploiting information about the target's current personal or professional situation (e. g. hobbies or work-related activities), the target can be tempted to open either infected attached files or visit suggested Web sites.

When open-source botnets are employed, the infection routines are generally self-developed or developed and shared within the community. When specialized botnets are used, the situation is similar, but for different reasons. Here, secrecy and often the environment in which the botnet is operating necessitate own developments. In construction kit-based botnets, infection vectors are usually supplied in the form of the already mentioned exploit kits.

The time required for the infection of hosts is difficult to estimate for all three classes of botnets. This also depends on the definition of "a sufficient number" of nodes which can be different for different purposes. When botnets are created for renting or selling them to third-parties [27], a common size of 10,000 hosts is bundled. For open-source and kit-based botnets, 10,000 hosts can often be infected within several hours to several days. In seldom cases, this can take more than a week. Specialized botnets often do not have the target of maximum infection speed as their purpose may not be financial gain but rather the accomplishment of long-term goals such as espionage. Thus, infection speed may not be of the utmost importance.

IV. RESOURCES REQUIRED FOR TACTICAL COUNTERMEASURES

Most of the common defensive techniques, such as firewalls, IDS, or anti-virus solutions, act on a local level. The locality is a problem when multiple targets are attacked that are managed by different entities, e. g. organizations with independent but cooperating branches. In addition, local measures can usually not prevent specific types of attacks, like DDoS attacks. A more sustainable and reliable way to counter such attacks is to conduct tactical countermeasures against the originating botnet.

In the following we will discuss the resources required to conduct different countermeasures that have the potential to take down the whole botnet. Two major types of countermeasures are considered. The first is classical countermeasures, which are rather moderate in their implications, but are very limited because of their dependence on the cooperation of other organizations. The second type is more aggressive countermeasures with global consequences which can be conducted by a single organization and are therefore, more suitable for a tactical take-down.

Each of the presented countermeasures is discussed with regard to the resources money, human resources and skill-level, cross-domain expertise required by those, time for conducting the countermeasure, sustainability, and possible legal or ethical constraints. Since many factors influence the different resources, no hard numbers are given but rather important relationships and estimations are explained.

A. Classical Countermeasures

1) C&C Server Takedown

If the location of a C&C server has been determined, it can theoretically be shut down or disconnected. This can be made difficult if redundant infrastructures spread multiple instances of the server all over the world, in particular hosting them with different providers. In addition to the main C&C endpoint(s), backup channels have to be identified, if the takedown is to be sustainable. If this has been achieved, sustainability is usually very high, especially for kit-based botnets for which details about the infrastructure are either freely available or can be purchased from security companies or malware intelligence (e. g. [28]). The same is true for open-source botnets, as source code analysis can easily reveal structural information. Specialized botnets, due to their stealthy nature, require significant effort in malware dissection by reverse engineering and forensics along with time and money to identify C&C endpoints and backup channels. .

Besides the required skills and money, cross-domain challenges like organizing cooperation with Internet service providers and local law enforcement authorities need to be faced. In an ideal case, the required time is in the vicinity of one hour. However, if lengthy analysis is needed and actions have to be coordinated with law enforcement in different countries, the entire process could take several months, if it is possible at all.

Legal constraints in some countries prohibit or complicate the takedown of C&C servers, enabling so-called bulletproof hosting requiring law enforcement intervention. In some countries, authorities and ISPs are reluctant to cooperate with security researchers or other security authorities. This is well-known and taken advantage of by botnet operators. Some ISPs notify customers if a site is about to be taken down and botnet operators can move the C&C server to another provider or a different country entirely.

2) DNS-based Countermeasures

If the C&C infrastructure of the botnet is based on DNS, then a classical countermeasure is deregistration of those domains required by the botnet. This has to be done in cooperation with the respective DNS registrars and was successful in several cases. A requirement for this countermeasure to be sustainable is that the botnet's C&C infrastructure relies solely on DNS mechanisms. If this requirement is met, DNS countermeasures are independent of the class of botnet, although C&C mechanisms tend to be more sophisticated in kit-based (Twitter-based selection of C&C server names in Torpig) and specialized (Kraken, Conficker) botnets.

Where money, skills and cross-domain knowledge is concerned, the main organizational challenge is the cooperation with the DNS registrars. These companies have no immediate benefit from such cooperation and often do so mainly because of the publicity effect. National and international law enforcement agencies also need to be coordinated with as there are legal issues to be considered. In the majority of cases, a court warrant needs to be obtained.

The time needed for this countermeasure to come into effect is affected by both the duration of the legal proceedings, i. e. to obtain the court warrant, and the time it takes for the DNS settings to be propagated to other servers. The latter can take from several minutes to several days, depending on the DNS time-to-live settings.
Already connected computers are not affected by this countermeasure; only newly connecting hosts performing a lookup receive the false information. Thus, the size of the botnet steadily decreases. The sustainability is very high.

B. *Proactive Countermeasures*

Beside the classical countermeasure, there are also more effective proactive countermeasures.

1) Response DDoS

If the locations of the C&C endpoints are known, a possibility is to launch a counter-DDoS attack to disable these endpoints. This is only possible if there is a single or limit number of C&C servers and would not work in a botnet relying on P2P infrastructure. A requirement for this is the availability of one or more machines for creating the traffic.
Financial resources are needed for the setup and operation of the traffic creation machines. This could, for example, be done by renting a competing botnet. According to [21], a DDoS botnet can be rented from 200 USD per 24 hours or 500 USD per month. Experiments conducted by an unnamed source have shown that a range of C&C servers can almost be shut down by DoS attacks from a single machine. This countermeasure is generally independent of the category of botnet being attacked. However, to determine the botnet's operating parameters, especially its C&C endpoints, can require extensive analysis. The resources in terms of skills, cross-domain activities and money required for this are comparable to those of the C&C server takedown described earlier.
The application of a counter-DDoS is possible practically instantly as soon as information about the C&C endpoints is available. However, the sustainability is negligible. The attacked botnet is disabled only as long as the counter-DDoS is executed. Also, the implications of launching an own DDoS attack need to be considered. It has to be ensured that legitimate services running in close proximity to the C&C endpoints are not adversely affected. In addition to that, DDoS attacks are illegal or even considered a hostile act in most countries.

2) Hack-Back

Another proactive countermeasure is hacking back, i.e. penetrating the C&C server and taking down the botnet from within. This requires the existence of a flaw in the C&C infrastructure which needs to be found and exploited. A team of highly skilled penetration specialists needs to be involved.
In open-source botnets, the C&C protocol can be easily discovered by analyzing the source code. Standard source code auditing tools can be used to find weaknesses in the code. Construction kit botnets are usually sold together with the C&C server, although it is typically in binary form. Therefore, reverse engineering

and binary code auditing skills are required. For specialized botnets it can be very difficult to obtain information about the C&C server. It is sometimes possible when using standard components with known vulnerabilities, e.g. specific Web servers. In all cases, analysts are required who are able to think outside box and identify non-obvious relationships between botnet components. Kit-based and specialized botnets require the highest reverse engineering skills.

The time required for such a hack-back differs among the different botnet classes. Because of the multitude of available code analysis tools, open-source botnets can often be hacked in a matter of minutes if a sufficient number of vulnerabilities exists, otherwise it is a matter of days depending on the complexity of the code. More time is required for kit-based botnets, since reverse engineering is needed most of the time. Because the server binary is available, offline and local stress tests can be performed. A minimum of several days can be expected, although the required time is more likely along the order of magnitude of weeks. Hacking of specialized botnets is very difficult. First the protocol has to be reverse engineered and possible weaknesses need to be derived. At least several weeks are needed for this.

Once access to the C&C server has been gained, diverse valuable information can be discovered. The installation of a root kit allows the complete control of the server machine and might even result in greater privileges than even the botherders have. However, in most countries it is illegal to gain access to computer systems of others without their knowledge. From an ethical point of view, hacking back is effectively fighting fire with fire.

3) Infiltration/Manipulation

Another proactive countermeasure is the infiltration of a botnet which might lead to the botnet being manipulated and/or disabled from within. This requires an in-depth understanding of the botnet's architecture and C&C protocol.

The skills needed vary for the different categories of botnets. Standard protocols, e. g. IRC and HTTP, can be automatically extracted, but especially for kit-based and specialized botnets, extensive reverse engineering skills are essential. Also, botnet domain knowledge coupled with out-of-the-box thinking is necessary to determine non-standard protocols. Cross-domain expertise is needed to identify and exploit weaknesses in the C&C protocol or architectures. Related fields in this case are communication protocol design, structured auditing as well as cryptography. Nevertheless, some manipulation vectors for standard protocols are well-known and can often be applied.

In terms of financial expenditure, analysis and monitoring environments need to be designed and built. Some organizations receive up to 100,000 malware samples per day. An investigation for the use of standard communication protocols takes place within a sandbox which has a minimum analysis time of 2 minutes. This requires around 140 machines running in parallel. Employing some heuristics allows the analysis to stop early. In addition to that, machines for monitoring are needed. Their number depends on the number of infiltrated botnets. Examples for existing frameworks for monitoring botnets are [17, 29].

The time required to infiltrate a botnet is difficult to estimate. A prerequisite is that malware samples are available for analysis. Their collection can already be a time-consuming task, especially if server-side polymorphism is used. Gaining an in-depth understanding of the botnet and its structures is also necessary. In case of standard protocols with standard manipulation vectors, a tactical takedown can be accomplished within minutes. The infiltration of botnets with non-standard protocols and the corresponding analyses can take up to several weeks, in lucky cases several days.

The sustainability of botnet infiltration is typically very high, provided it is not pursued too aggressively. For example, the aggressive monitoring of Storm by researchers was obvious to the botnet operators. If manipulations are made on the C&C server, they can be detected most of the time. To be truly effective, sudden strikes are essential.

The legal aspects of botnet infiltration still need to be investigated. From an ethical point of view, only the botnet's operation is interfered with. However, third-party data might be obtained or manipulated as a consequence, especially if the C&C is hosted on a hijacked system and depending on the architecture.

4) BGP Blackholing

Another possibility is the redirecting of botnet-related traffic, so-called sinkholing. The redirected traffic can simply be discarded or analyzed further to gather more information about infected machines. Resources with regard to money, skills and cross-domain knowledge are similar to those of regular C&C server takedowns. The processes can mostly be fully automated. However, the existence of backup channels for C&C processes can be challenging. Once sufficient information about the botnet and its structures is available, the C&C endpoints can be inserted into BGP feeds within a few seconds, although their propagation can take several minutes.

V. SUMMARY AND OUTLOOK

In this paper we have discussed the resources required for setting up and taking down botnets. In order to structure this we have categorized botnets into three groups: completely open-source botnets or those that use open-source components, construction kit-based botnets which are normally for sale, and specialized botnets tailored to a very specific task.

In general, kit-based botnets are the easiest to setup and operate since they were designed with user-friendliness in mind. When setting up an open-source botnet, basic software development skills are required which can be obtained in a matter of hours or days. Challenges can often be overcome by taking advantage of community support. This community support is missing for specialized botnets, often due to secrecy requirements.

Classical countermeasures are often inadequate when faced with intricate botnet structures and protocols. Proactive countermeasures are much better suited to deal with the botnet threat. Sufficient funds, time and development expertise in the area of malware analysis and reverse engineering are the most important requirements.

There is an increase in the amount of the respective required resources from open-source botnets through kit-based botnets to the specialized variants.
Currently, botnet operators are ahead in the arms race with security researchers, the anti-virus industry and law enforcement agencies. The currently performed anti-botnet activities are not as aggressive as they could be. This is partially due to lack of resources, the fear of legal consequences or uncoordinated efforts but also sometimes because of the fear of an intensifying arms race. Another reason is that monetary losses in the often targeted financial industry are still relatively moderate. However, studies show that there is a steady increase in the amounts lost due to credit card fraud, extortion and other botnet-related crimes. Thus, with a corresponding increase in funds for anti-botnet activities, it stands to reason that there will be more offensive botnet takedown attempts in the not-too-distant future, despite the fact that this would spark the feared arms race.

REFERENCES

[1] Symantec. Press release. Available online: http://www.symantec.com/about/news/release/article.jsp?prid=20090910_01, accessed February 2011.

[2] J. Nazario. *Politically Motivated Denial of Service Attacks*. In: C. Czosseck, K. Geers (Eds.), “The Virtual Battlefield: Perspectives on Cyber Warfare”, IOS Press, 2009.

[3] Shadowserver. *60-Day Virus-Stats*. Available online: http://www.shadowserver.org/wiki/pmwiki.php/Stats/Virus60-DayStats, accessed February 2011.

[4] F. Leder, T. Werner, P. Martini. *Proactive Botnet Countermeasures – An Offensive Approach*. In: C. Czosseck, K. Geers (Eds.), “The Virtual Battlefield: Perspectives on Cyber Warfare”, IOS Press, 2009.

[5] G. Klein, F. Leder, “Latest trends in botnet development and defense”, Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia, 2010.

[6] *Metasploit - Penetration Testing Resources*. Available online: http://www.metasploit.com, accessed February 2011.

[7] *OpenSSL: The Open-Source Toolkit for SSL/TLS*. Available online: http://www.openssl.org, accessed February 2011.

[8] R. L. Rivest et al. *The MD6 hash function – A proposal to NIST for SHA-3*. Technical Report. Massachusetts Institute of Technology, Cambridge, MA, USA, April 2009.

[9] J. Gailly, M. Adler. *zlib Compression Library*. Available online: http://www.zlib.net, accessed November 2010.

[10] D. Fisher. *Storm, Nugache lead dangerous new botnet barrage*. Available online: http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1286808,00.html, accessed December 2010.

[11] *VMProtect homepage*. Available online: http://vmpsoft.com, accessed November 2010.

[12] N. Villeneuve. *Tracking GhostNet: Investigating a Cyber Espionage Network*. Available online: http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network, accessed February 2011.

[13] N. Falliere, L. O. Murchu, E. Chien. *W32.Stuxnet Dossier*. November 2010. Available online: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf, accessed February 2011.

[14] PC Plus. *Botnets Explained*. Available online: http://pcplus.techradar.com/feature/features/botnets-explained-30-09-10, accessed November 2010.

[15] Stevens, K., Jackson, D. *ZeuS Banking Trojan Report*. Available online: http://www.secureworks.com/research/threats/zeus, accessed December 2010.

[16] M86 Security Labs. *Web Exploits: There's an App for That*. Technical Report. Available online: http://www.m86security.com/documents/pdfs/security_labs/m86_web_exploits_report.pdf, accessed November 2010.

[17] Marco Cremonini and Marco Riccardi. *The Dorothy Project: An Open Botnet Analysis Framework for Automatic Tracking and Activity Visualization*. In *Proceedings* of the 2009 European Conference on Computer Network Defense (EC2ND '09). IEEE Computer Society, Washington, DC, USA, 52-54

[18] Kleissner, P. *AV Tracker homepage*. Available online: http://www.avtracker.info, accessed November 2010.

[19] Higgins, K. J. *Zeus Attackers Deploy Honeypot Against Researchers, Competitors*. Available online: http://www.darkreading.com/insider-threat/167801100/security/attacks-breaches/228200070/index.html, accessed December 2010.

[20] Websense, Inc. *Websense 2010 Threat Report*. Technical Report. Available online: http://www.websense.com/content/threat-report-2010-introduction.aspx, accessed November 2010.

[21] Damballa. *Want to rent an 80-120k DDoS Botnet?*. Available online: http://blog.damballa.com/?p=330, accessed February 2011.

[22] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir. *A survey of botnet technology and defenses*. In Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security, pages 299–304, Washington, DC, USA, 2009. IEEE Computer Society.

[23] P. Bächer, T. Holz, M. Kötter, and G. Wicherski. *Know your enemy: Tracking botnets*. Honeynet Project KYE series, 2007.

[24] F. Leder and T.Werner. *Don't do this at home - owning botnets*. In T2 information security conference, Helsinki, Finland, 2009.

[25] McAfee, Whitepaper, *Global Energy Cyberattacks: "Night Dragon"*, Version 1.4, February 10, 2011, http://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.pdf , accessed February 2011

[26] D. Fisher. *Storm, nugache lead dangerous new botnet barrage*. http://searchsecurity.techtarget.com/news /article/0,289142,sid14_gci1286808,00.html, , accessed February 2011

[27] Spencer Kelly, BBC, *Gaining access to a hacker's world*, 13 March, 2009, http://news.bbc.co.uk/2/hi/programmes /click_online/7938201.stm , accessed February 2011

[28] Abuse.ch, *Zeus Tracker*, https://zeustracker.abuse.ch/ , accessed February 2011

[29] G. Wicherski, *botsnoopd - Efficiently Spying on Botnets*, GovCert Symposium, September 16, 2008, Rotterdam, NL