Measuring Botnet Prevalence: Malice Value

Christian Nordlohne

nordlohne@internet-sicherheit.de

Institute for Internet Security,

University of Applied Sciences Gelsenkirchen,

Neidenburger Str. 43, 45877 Gelsenkirchen, Germany

January 7, 2015

Abstract

Fighting malware, and paticulary botnets is a challenging task. Prioritization of botnets to fight first is often led by personal intuition or the customers needs. Within this paper we show two different methods for measuring botnet prevalence. The first method takes only the size of a botnet into account. Using this method we show that it is prone to artefacts of the used database and not reliable. The second method is our developed malice value metric, which is a transparent, comparable and comprehensible way to measure the prevalence of malware and botnets. Comparing the values for 15 botnets the malice value produces numbers which match the expectations.

1 Introduction

Malware, and particulary botnets, are an everlasting threat to computer security. Over the years botnets evolved from specially crafted IRC clients like the Virut family [39] to very sophisticated pieces of software that uses state-of-the-art encryption for communication [11]. There are many questions to be answered if you want to fight malware and botnets. One of these question to be answered is: „What are today's most prevalent botnets?“, or in other words: „Which botnet should we fight first?“. The answers to these questions are not given as easy as it may seem. With this work we would like to show two different approaches to answering this question. We point out the weakness of only considering the size of a botnet as a prevalence factor and instead present a new way to estimate the real impact of different botnets. We present a metric for the malice value which describes the “badness” of a piece of malware, in specific: of a botnet.

1.1 Motivation

How to measure the prevalence of botnet families? To measure something, a scale is needed that assigns values to the different malware families. So what should the scale be? How does one assign values to rate malware families?

2 First Try: Botnet prevalence by size

The first step to measure the prevalence of botnets is to identify the relevant factors and scale and to determine the ranking. A trivial attempt would be to measure the prevalence of a given botnet is simple: The bigger the botnet the more prevalent it is. The idea behind this is simple: If a spam botnet has many infected hosts, it could send more spam and therefore could do more damage. The same applies to any other malicious activity a malware could utilize (DDoS, Data stealing, encryption of personal data etc.). Following this, the size of a botnet is the scale for prevalence. The botnet size has been subject to investigations many times and all of them have their specific problems[18]. Rossow et al.[33] measured the size of P2P botnets using crawling peer lists and sensor nodes. Their results indicate that some botnets contain more than a million infected hosts. For centralized botnets the sizes are more difficult to measure. One can get the most accurate number of zombies by taking over a centralized botnet[36]. Even if a centralized botnet is infiltrated its actual size and number of active bots is difficult to measure as a botnet could use multiple linked C&C servers [14]. Step two is to assign numbers to the different malware families and botnets specifically to create a ranking. For this we decided to take different data sources, which should be as comprehensive as possible, to get the best overview. Many security companies and AV vendors present their findings about malware and botnets in public reports. In this first trivial attempt we use these reports as database.

2.1 Artefacts of the database

The main problem with these databases is that you cannot get any accurate information about them. For example in Microsoft's SIR[24] the source of data is reported to be "over 600 million computers worldwide", whereas other vendors do not give a number[9, 40]. The naming of malware is a problem too. Different vendors have different names or "labels" for the same piece of malware. For example: The Rimecud (Microsoft) botnet is also known as Palevo, Mariposa, Butterfly and Pilluz. This results in manual work to do: one has to take care of this by comparing the names and looking for synonyms. Withinn these reports only "disinfected" hosts are counted, so it is more like a size that describes the "number of hosts that had been infected once". Last but not least many of the reported malware in the "Top 10 in 2013" are not botnets at all (for example a signature like "Keygen" or "autorun") and therefore could not be used.

2.2 Databases

The trivial attempt relies on the following sources:

- Symantec Security Threat Report 2013[38]
- Microsoft SIR 14[24]
- McAfee Threats Report Fourth Quarter 2012[22]
- eset Global Threat Trends April 2013[9]
- TrendMicro 3Q 2012 Security Roundup[40]
- F-Secure Threat Report H2 2012[10]
- FortiGuard Threat Report (website)[13]

The table below lists the source, its database size in terms of contributing sensors and how much the top 10 malware accounts for in this database. For example: In Symantec's Security Threat Report 2013 the top 10 malware is responsible for 40% of all infections. This was reported by "[...]well over 133 Mio. clients, servers and gateway systems[...]" [38]

Source	**Size of Database**	**Top 10 is % of total**
Symantec	133 Mio.	40
Microsoft	600 Mio.	54
McAffee	„millions“	n/a
ESET	n/a	17
TrendMicro	n/a	n/a
F-Secure	n/a	n/a
Fortiguard	1 Mio.	5

This table also illustrates the artefacts using such databases as stated above. The total number of malware family names were 48 and of those 43 were unique. Note that not all reports count the “Top 10” of all infections, therefore the total number is not 70, as seven reports each with 10 (as in Top10) malware listed.

2.3 Method

To correlate these data we have chosen a simple approach. We count the occurrence of a specific malware among all lists and divide this by the average position in all lists. This would be the rating for the final position.

$$rat = \frac{\text{present in X lists}}{\text{average position in lists}}$$

For example: Malware A is in three out of seven lists and is rated in list one at position two, in list two at position one and in list three at position three. Its average position in lists is $\frac{1+2+3}{3} = 2$. To determine its final position (rat) we take the number of lists malware A is present in (three) into account. The final rating for malware A is $rat = \frac{3}{2} = 1.5$.

2.4 Result

Taking the ratings using the method above we create the final list by sorting it from lowest rating to highest. Using this, and by removing all members of the list that are not known to be botnets, the result is shown in the table below.

Position	**Botnet**	number of lists	average position	rating
1	ZeroAccess	4	3	1.33
2	Conficker	5	3.8	1.31
3	Dorkbot	4	5.5	0.72
4	Sality	3	4.33	0.69
5	Ramnit	2	4	0.5
6	Zeus	2	6	0.33

Botnet are named as the "most common" used for them as known to the author.

2.5 Discussion

The main problem resulting from this approach are database artifacts screwing the results (see table above). But if you consider the artefacts of the databases which where used, the list makes perfect sense. ZeroAccess is a banking Trojan based on the Zeus toolkit and a very big botnet with an estimated 350,000 infected hosts active[33]. Since ZeroAccess is a big and well known botnet, anti virus vendors are trying hard to produce signatures and remove it from the infected hosts of their customers. Conficker on the other hand is an "old" inactive botnet, but educated guesses say it may still be spread on hundreds of thousand hosts. There are signatures for all versions of Conficker and almost every popular anti virus software will remove it. Position two for Conficker could be explained by the fact that more and more computers are online as the availability of high bandwidth grows. These computers may use old OS installations and therefore not have all the security updates available which prevent the exploit that Conficker uses to infect systems.

2.6 Conclusion

To measure the prevalence of botnets using this method produces a result which could be explained well but is not sophisticated. The rating this methods produces could not be taken as an advice which botnet we should fight first, because if you look at Conficker on position two, for example, every security expert would argue that it is no longer active, the exploit does not work anymore and as soon as you install AV software you get rid of it. So following this conclusion there must be another way to measure the prevalence of botnets. We want to take more factors into account for the rating and not have the artefacts of our database have such a huge negative impact on the results.

3 Creating a metric to measure botnet prevalence

3.1 Motivation

As shown in the section above a more sophisticated approach to measure botnet prevalence is needed. First of all the new method must fulfil the following requirements:

- It must rely on **multible factors**
- It must be **transparency**
- It must be **comparability**
- It must be **comprehensiveness**

The previous method where we used the number of reported infections as the size of a botnet to describe its prevalence was prone to multiple artifacts as shown previously. To use multiple factors to describe a botnets prevalence is necessary. Transparency and comprehensiveness are needed because the results the new metric produces should be reproducible and show exactly why a specific result is created. Comparability is needed as we want to rank different botnets and make a statement about which botnet is *more malicious* then others.

3.2 Factors for botnet prevalence

To create a metric which satisfies this requirements the first thing to do is to identify the factors. There are several factors that could be considered and they will be discussed below.

3.2.1 Size

Size is definitely a factor which describes the prevalence of a botnet. But the question is: how big should its impact on the result be? Let's consider two botnets A and B. Botnet A is a spam botnet with 1 million hosts. Botnet B is 1% the size (10,000) but it is used for stealing credit card numbers and performing banking fraud. Considering the damage potential of those botnets leads to the conclusion, that botnet A is more "harmless" than B from the users point of view. This is because there are many reliable techniques in place to fight spam. A sophisticated banking trojan could do way more damage to end users. The next problem with size is the potential wide range of numbers. If we want to assign a number for the prevalence, and the size is used in the metric with its "actual" number like 525,456 active bots, the magnitude would be something between 1 and some million. So size, representing the infected machines of this botnet, is definitely a factor, but should be handled carefully and scaled down in some way.

3.2.2 Infection

The way and effort a botnet takes to infect a new system is the next factor to be considered. If you have a bot that uses state of the art zero-day exploits on websites, infects PDF files and spreads them via e-mail, it is an indicator that the "bad guys" have put a lot of work into making their botnet work and spread out. So you could consider such a botnet more prevalent than one which uses old exploits, which would only work if the target system has an unpatched OS and software. We will name this factor infection in the final metric. This factor will take into account the different techniques used, like unpatched vulnerabilities and spreading via network and assign values to them.

3.2.3 Stealth

Considering the potential damage a botnet could cause, another factor to be considered is the answer to our next question: "How long could the bot hide on a system and do its work without the user, or AV software noticing it?" For example: If a company gets infected with a botnet that steals blueprints, this question is critical to estimate the potential damage. Has all data been stolen or only partially? The risk of data being stolen increases with the duration of the bot being active. From the viewpoint of an end user this could be the following scenario: The PC gets infected with a botnet that targets among others eBay and Paypal credentials. The malware scans the computer for those credentials or wait for them by installing a keylogger. If the user only log in to ebay on weekends (and has not stored his credentials on hard drive) and the malware is detected by anti-virus before the weekend the account may not be compromised. The duration for which a botnet or malware in general stays hidden on a system is one factor to be taken into account. We name this factor stealth.

3.2.4 Activity

Today's botnets start their task as soon as they infect a new system: They connect to their Command and Control (C&C) server and wait for commands. As example, IRC botnets like Dorkbot[25] sends and receives "PING PONG" messages every 30 seconds. On the other hand there are botnets like Conficker: these are botnets that could do a lot of damage, as they have a large number of infected machines, but are no longer active. This must be considered in the metric. Another advantage of taking this factor into account is, that we can make a statement about how much the prevalence of a botnet decreases over time if it is taken offline or is no longer active. Summarizing activity is the factor that describes the minimum time a given botnet has been active.

3.2.5 Resilience

The resilience of a botnet has a huge impact on the effort one must make to bring it down. Resilience will represent the techniques used by a botnet to make it more resilient like using Peer to Peer (P2P) or encrypted Command and Control

messages. A centralized IRC botnet could be taken offline by locating the server and cut the power line. If it uses fast flux techniques to change domain names[27] or a Domain Generation Algorithm (DGA) to produce hundreds of possible C&C domains to contact the more complicated it is to take it offline or interfere its work. Each technique a botmaster uses makes the botnet more resilient and more dangerous. These techniques are used as parameters to calculate values and will add up to a final score for this factor. The resilience of a botnet therefore is another factor we will use in the metric.

3.2.6 Target

The intended target of a botnet is another critical factor to measure its prevalence. Many botnets target end users with potentially unpatched and outdated operating systems and software. But there are botnets and malware, like Stuxnet and Flame, that specifically target parts of a country's industry or critical infrastructure. The effort to infect those systems are much higher, as corporations and governments use their own security infrastructure like IDS, IPS and whole security departments, which work all day to prevent malware from infecting the infrastructure. So a botnet that targets critical environments should be considered more dangerous.

3.2.7 Damage

The actual damage potential of a botnet is hard to estimate. Security companies do provide numbers like "Spam costs 100 million US Dollar a year" but assigning a financial damage value to a specific botnet is a challenging task. It may be possible for banking Trojans but can if at all only be roughly estimated when measuring the damage done by botnets that target companies or critical infrastructure. Therefore we will not use this factor in our final metric.

3.3 Creating a model for the factors

To meet the requirement of comprehensibility, every identified factor must have a set of characteristics with each one assigned a value to it. We provide a default value for every factor to make the metric work in an environment with less information coverage. The following numbers assigned for the different characteristics should be considered as an example, as there is a lot of place for discussions which characteristic should be rated higher or lower. The values assigned to the different characteristics of a factor will be named a *model* for the metric.

3.3.1 Size

The size of a botnet can be described as number between 1 and infinity[1] and there is by definition a value we can use. The metric must limit to an intervall

[1] practical values are natural numbers

that affect the prevalence but are not the only crucial factor. We consider a default botnet to have a size of about 100,000 infected machines[2].

	Value
Minimum	1
Maximum	Infinity
Default	100,000

3.3.2 Infection

The ways a malware and therefore a bot infects a computer should represent the effort undertaken by the "bad guys" and their techniques used. As malware could use multiple ways to infect a system, like using unpatched vulnerabilities in web browsers or being downloaded by other malware we assign each technique a value and add them all together to get the final value for this factor.

Technique	**Value**
uses unknown vulnerability	0.325
uses unpatched vulnerability	0.175
spreads automatically via network	0.05
needs to be executed by user	0.05
uses removable device (eg. USB)	0.05
not detected by any AV	0.1
not detected by IDS/IPS	0.1
spreads by websites	0.05
downloaded via other malware	0.05
uses email/documents	0.05

We set the default value for this factor to 0.325. This represents a botnet using an unpatched vulnerability (0.175), spreads via infected websites (0.05) and is not detected by anti-virus software (0.1). As our research shows this is true for an average botnet. The two most dangerous techniques a malware could use to infect a system are unknown and/or unpatched vulnerabilities in OS and other software. These two characteristics add up to half of the maximum risk value of 1 in this model. Using unknown vulnerabilities should be considered way more dangerous than using known vulnerabilities it is rated twice as high. This factor takes into account the possibility that malware may change over time. Malware gets detected by AV and IDS/IPS as signatures are created and distributed to end users. This could be used to create values which represent the infection factor at the time the malware is discovered and the time AV signatures are available and maybe patches for the former unknown vulnerability are available. This is related to the Common Vulnerability Scoring System

[2]This default value is reached by taking the rounded average value of estimated infected hosts of about 32 known botnets

presented by Mell et al.[30] but as our metric rates the malware using the vulnerability i slightly different aproach is needed.

	Value
Minimum	0
Maximum	1
Default	0.325

3.3.3 Stealth

The stealth factor represents the dangers created by a botnet remaining undetected on a system. For example: A keylogger could gather more passwords and credentials maybe for more then one user if it is undetected. Another good example is Stuxnet, which could unfold its full damage potential by disturbing the centrifuges over a long period of time. We choose a quadratic growth of this factor for our model to fit this circumstance best. We want to map the days of a year (365) to the values from 0.01 to 1 with quadratic growth. To achieve this, we square the value and multiply this with 365 to get the threshold of time. To get better thresholds to work with, we will not take the exact values but instead use whole days/weeks/month which are close to the exact value. The table below shows the values for this factor in our model.

Scale	**Square**	**Days**	**Days hidden (upper bound)**	**Value**
0.01	0.0001	0.0365	1 days	0.0001
0.1	0.01	3.65	3 days	0.01
0.2	0.04	14.6	2 weeks	0.04
0.3	0.09	32.85	1 month	0.09
0.4	0.16	58.4	2 month	0.16
0.5	0.25	91.25	3 month	0.25
0.6	0.36	131.4	4 month	0.36
0.7	0.49	178.85	6 month	0.49
0.8	0.64	233.6	7 month	0.64
0.9	0.81	295.65	9 month	0.81
1	1	365	>=1 year	1

The default value for this factor is 0.04. This represents a botnet staying hidden on a system for a time no longer than two weeks, which is a rough guess by the author taking into account the time it takes to analyse a new malware, create a signature and make it available for AV software. This is similar to the "not detected by AV" characteristic from the factor infection, but at that factor it is rated towards the actual mode of infection. If a malware is detected by AV-Software and you get infected, nonetheless you either have no AV-software installed or have not installed the latest update. This fact is represented by the infection factor. The stealth factor therefore represents the

time a specific bot that infected your system is neither detected by IDS, AV-software nor manual investigation after observing strange behaviour in your network or on the infected system.

	Value
Minimum	0.0001
Maximum	1
Default	0.04

3.3.4 Activity

Representing if a botnet is active at all or have not been active for a given time period this value should decrease with the time a botnet has been inactive. The activity shall decrease in the same way as the stealth values. Therefore we take the quadratic values in this model like we did for stealth but the time is inversed.

Last time active (lower bound)	**Value**
> 1 year	0.0001
> 9 month	0.01
> 7 month	0.04
> 6 month	0.09
> 4 month	0.16
> 3 month	0.25
> month	0.36
> 1 month	0.49
> 2 weeks	0.64
> 3 days	0.81
Today	1

Analysing a database of over 500,000 samples we assume in this model the average botnet receives or asks for commands several times a day and therefore the default value for this factor is 1. A botnet is considered “active” if it sends commands.

	Value
Minimum	0.0001
Maximum	1
Default	1

3.3.5 Resilience

Like malware uses different ways to infect a system, the resilience of a botnet can utilize a variety of tactics to increase its resilience. The characteristics and

their values used in our model are described in the following table. P2P is considered to be the most resilient resilience a botnet could use, as there is no single point of failure. This technique is assigned the highest value, which is twice as much as a centralized C&C resilience would score. All characteristics and values in this model are described in the table below.

Technique used	**Value**
centralized C&C	0.2
P2P C&C	0.4
Fast Flux	0.05
Double Fast Flux	0.1
DGA	0.1
updating possible	0.05
encrypted C&C	0.1

The default value for the factor *resilience* is calculated like this: a botnet that uses a centralized C&C server (0.2) employs h a DGA (0.1), encrypts its C&C commands (0.1), and has update functionality (0.05). This scores 0.45 points in our model.

	Value
Minimum	0
Maximum	1
Default	0.45

3.3.6 Target

The final factor we integrate into our metric is the target of a botnet. There are many different targets like end users, companies, and critical infrastructures. To represent the differences we assign a subset of all end users the lowest value (like the botnet targets a specific country) and critical infrastructure the highest, as a botnet affecting the infrastructure of a country could have catastrophic impact on society and economy. Our model uses the following numbers for the described characteristics.

Target	Value
Limited users	1.01
Many users	1.1
All users	1.2
SME	1.3
Mid-sized companies	1.4 1.5
Big companies	1.6
Banks	1.7
Military	1.8
Critical Infrastructure	1.9 2

As the average botnets targets end users to make money (stealing credentials etc.) the default value for *target* is 1.2.

	Value
Minimum	1.01
Maximum	2
Default	1.2

3.3.7 Creating the metric

Since our metric considers all those factors, it represents more than the just prevalence. We can think of it as indicator for the “badness” or “level of dangerousness” of a botnet or malware. Stick to the term "malware" (malicious software) we name the value produced by the metric the **malice value** of a malware/botnet. As stated above, the malice value should be scaled down to make the malice value meet our requirement for being comparable. The size of botnet may vary between 1 and infinity. To scale down large numbers we use logarithm. In the following explanation *size* means the *footprint* or number of infected systems of a botnet. To produce values above one and in a scope that has not the highest impact of all to the final value we normalize the size to *size'* as follows:

$$size' = 1 + \frac{log_{10}(size + 10)}{10}$$

Some further explanation is needed for this part. First of all the term

$$log_{10}(size + 10)$$

As described above we want to use the logarithmic with base 10 to normalize the actual number of infected systems. To avoid the problems of $\log_{10}(1) = 0$ we add a static number to the actual number of bots. The effect of this adding gets less in higher numbers of infected systems as shown in the table below, which compares the use of $\log_{10}(size)$ and $\log_{10}(size + 10)$.

size	$\log_{10}(size)$	$\log_{10}(size+10)$
1	0	1.04
10	1	1.30
100	2	2.00
1000	3	3.00
10000	4	4.00
100000	5	5.00
1000000	6	6.00

To push this factor to a value range which is more in line with the value range of the other factors we do some further normalizing shown in the table below. Our goal is to eventually normalize to a value between one and two[3].

size	$log_{10}(size+10)/10$	$1+\frac{log_{10}(size+10)}{10}$
1	0.10	1.10
10	0.13	1.13
100	0.20	1.20
1000	0.30	1.30
10000	0.40	1.40
100000	0.50	1.50
1000000	0.60	1.60

This leaves us with the size factor between 1.10 (for a botnet with one bot) and 2 for botnets with 10 billion bots as the table below indicates.

size	**size'**
1	1.10
10	1.13
100	1.2
1,000	1.3
10,000	1.4
100,000	1.5
1,000,000	1.6
10,000,000	1.7
100,000,000	1.8
1,000,000,000	1.9
10,000,000,000	2

We will now scale *size'* with the factor stealth as they directly correlate with the potential of a botnet to do damage and therefore create the first part of the malice value:

$$m_1 = stealth * size'$$

[3]as this factor would be the base for an exponent a value smaller then one would result in decreasing values

The next factor directly affecting these two is the target of a malware. As this is a crucial factor affecting other factors we give it a very high impact on malice value and use it as an exponent:

$$m_2 = (stealth * size')^{target}$$

The remaining factors are directly affected by *target*, but not as strong as the first three. Therefore we add these factors and scale them by *target*. Summarized our final metric for the **malice value** of a botnet/malware is:

$$malice\ value = tar * (res + inf + act) + (stealth * size')^{tar}$$

Where *tar* denotes target, *res* resilience, *inf* infection and *act* activity. Malice value using the *model* for the factors described above results in a number between 0.00041 and 10 when using the minimum and maximum value for every factor except size. As infinity would not be a very useful upper bound, we use 10,000,000,000 infected machines as a fixed maximum. The probability for this value is very low, as it would represent a botnet with more than double the times infected hosts then there are IPv4 addresses. This is for sure a fictional value, but can be used to create an upper bound for malice value in this case. The table below shows the factors, their value and the resulting malice value using minimum, maximum, and default values in this model.

minimal	**maximal**	**default values**	**Factor**
0.0001	1	0.325	infection
0.0001	1	0.04	stealth
1	10,000,000,000	100,000	size
1	2	1.2	target
0.0001	1	1	activity
0.0001	1	0.45	resilience
0.00041	10.61	2.16	malice value

3.4 Results

Using the metric to determine malice value and the models described above to assign values to the characteristics for 15 known botnets we get the following results.

Botnet	**infection**	**stealth**	**size**	**target**	**activity**	**resilience**	**malice value**
Conficker[12, 31]	0.6	0.04	3,000,000	1.2	1	0.35	2.37
Conficker (inactive)	0.275	0.0001	2,000,000	1.2	0.0001	0.35	0.75
Waldec[1, 21, 37, 16]	0.15	0.04	12,000	1.2	1	0.55	2.07
Gameover Zeus[7, 4]	0.6	0.04	250,000	1.2	1	0.65	2.73
Gameover Zeus (inactive)	0.275	0.04	250,000	1.2	0.25	0.65	1.41
Virut[39]	0.05	0.04	300,000	1.2	1	0.35	1.71
Khelios[6, 33]	0.05	0.04	110,000	1.2	1	0.55	1.95
Citadel[23]	0.6	0.04	5,000,000	1.2	1	0.45	2.49
Koobface[26, 8]	0.15	0.04	500,000	1.2	1	0.35	1.83
Rimecud/Palevo/Maripose/Butterfly[19, 5]	0.05	0.04	800,000	1.2	1	0.35	1.71
Pushdo/Cutwail[17]	0.1	0.04	1,500,000	1.2	1	0.35	1.77
SpyEye[34, 3]	0.15	0.04	1,400,000	1.2	1	0.25	1.71
Tdss/ Tdl / Alureon[20, 2]	0.15	0.04	90,330	1.2	1	0.75	2.31
Stuxnet[29]	0.8	1	24	2	1	0.65	6.22
Flame[35]	0.6	1	10	2	1	0.45	5.37
ZeroAccess[15, 33]	0.1	0.04	350,000	1.2	1	0.75	2.25
Sality[28]	0.05	0.04	100,000	1.2	1	0.7	2.13

For each botnet sources where queried to determine the characteristics and using the values from our model for the factors.

4 Discussion

We have created a metric to determine the malice value of a malware/botnet as comprehensible and comparable as possible. We evaluated and introduced additional factors to the size of a botnet, as this has proven to be too vulnerable to artefacts of the used database and other facts. Looking at the results for some different, known botnets of today and the past, we could observe that the metric produces values that are not contradictory. Let's look at the Gameover Zeus botnet for example. This botnet has been taken offline by a coordinated strike from the FBI and several security researchers around the world . The time it was active it scores with a malice value of 2.73. Gameover Zeus represents a high sophisticated piece of malware that uses many advanced techniques and has hundreds of thousand infected machines. Comparing it to the time after its takedown the malice value drops down to 1.11 due to being inactive and security patches being available to fix the used exploits. Another example is the Stuxnet botnet. This botnet had a completely different purpose then Gameover Zeus as it has targeted the infrastructure of the Iran. It too uses many advanced techniques and has been hidden on the systems for a long time. The actual damage it has done can only be estimated, but by scoring with a malice value of 6.22 it is in a very different range representing the fact that this is actually an Advances Persistent Threat (APT) and therefore way more dangerous than malware that tries to infect as many user machines as possible. Last but not least comparing the results using the two different methods which are described here fixes weaknesses of the first method which only uses the reported infection from AV vendors. If we use the malice value to rearrange the ranking created by the trivial approach lead to the results in the table below.

Ranking Method 1	**New ranking**	**malice Value**
ZeroAccess	ZeroAccess	2.25
Conficker	Sality	2.13
Dorkbot	Ramnit	1.95
Sality	Dorkbot	1.70
Ramnit	Zeus (inactive)	1.41
Zeus	Conficker	0.75

As shown in the table, the botnets which are known to be inactive fall down to the bottom of the list. Ramnit, Dorkbot, the inactive Zeus and the Conficker botnet would even not be in a new 'Top6' list if the malice value would be used to create a ranking.

5 Conclusion

To rate the "badness" of malware is nothing new, various security companies nowadays are doing the same. Our approach to a metric for the malice value we presented in this document is different. The metric fulfils the requirements of

being transparent, comparable and comprehensible. It works best if you have as much a-priori information as possible, but works with pre-defined default values for the different factors - which meet the requirements, too. To compute the malice values for 15 well known botnets we created a model that assign numbers to the characteristics for the different factors in the metric. The numbers for the model were taken from personal experience of the author analysing malware and botnets the past three years. By various examples the results of the metric match with the authors personal estimations.

6 Future work

The values used in a model for creating the malice value should constantly be updated to follow the changes and new dangers of malware. This could be achieved by monitoring the development and analysis of new malware to identify shifting priorities in any factor. A detailed mathematical analysis of the presented formula could reveal if any of the described and used factors are not as significant as assumed and could therefore be ignored.

7 Related work

The only other work known that tries to measure malware, is the Malware Rating System presented by Bagnall and French[32]. Our malice value metric is a different approach, as it takes into account a higher number of different factors. The Common Vulnerability Scoring System be Mell et al[30] focuses on the vulnerability itself where our malice value focuses on the malware.

8 Acknowledgements

I thank Christian Dietrich and Christian Rossow and my colleagues at if(is) for their helpful discussions and comments. This paper refers a work conducted within the pilot project ACDC (project no. 325188) , started in February 2013 and co-funded by the European Commission under the Information and Communication Technologies (ICT) theme of the Competitiveness and Innovation framework programme (CIP).

References

[1] abuse.ch. Waledac: Ein kleiner ueberblick, January 2009. `http://www.abuse.ch/?p=946`.

[2] abuse.ch. How big is big? some botnet statistics, May 2011. `http://www.abuse.ch/?p=3294`.

[3] Aditya Sood and Richard Enbody. Spying on spyeye. http://conference.hitb.org/hitbsecconf2011ams/materials/D2T3%20-%20Aditya%20K%20Sood%20-%20Spying%20on%20SpyEye.pdf.

[4] Dennis Andriesse, Christian Rossow, Brett Stone-Gross, Daniel Plohmann, and Herbert Bos. Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus. In *Proceedings of the 8th IEEE International Conference on Malicious and Unwanted Software (MALWARE'13)*, Fajardo, Puerto Rico, USA, October 2013. IEEE Computer Society.

[5] Bharat Jogi. Analysis: Malware win32/rimecud.b, May 2011. https://community.qualys.com/blogs/securitylabs/2011/05/09/analysis-malware-win32rimecudb.

[6] Brian Krebs. Researchers clobber khelios spam botnet, March 2012. http://krebsonsecurity.com/2012/03/researchers-clobber-khelios-spam-botnet/.

[7] Cert Polska. Zeus-p2p monitoring and analysis, June 2013. http://www.cert.pl/PDF/2013-06-p2p-rap_en.pdf.

[8] Emil Protalinski. Koobface gang pulls server after facebook exposes hackers, January 2012. http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=153670.

[9] eset. Global threat report, April 2013.

[10] F-Secure. Threat report h2 2012, 2012.

[11] Fedor Sinitsyn. A new generation of ransomware, July 2014. "http://securelist.com/analysis/publications/64608/a-new-generation-of-ransomware/".

[12] Felix Leder and Tillmann Werner. Know your enemy: Containing conficker, April 2009.

[13] FortiGuard. Fortiguard threat report, April 2013. http://www.fortiguard.com/fortiguard_labs_monthly/fortiguard-labs-monthly-2013-04/.

[14] Felix C. Freiling, Thorsten Holz, and Georg Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. In *In Proceedings of 10 th European Symposium on Research in Computer Security, ESORICS*, pages 319–335, 2005.

[15] James Wyke. The zeroaccess botnet – mining and fraud for massive financial gain, September 2012.

[16] Jonell Baltazar and Joey Costoya and Ryan Flores. Infiltrating waledac botnet's covert operations, September 2012.

[17] Kyle Yang and Derek Manky. Pushdo and cutwail: Enhancing command and control. http://www.fortiguard.com/legacy/analysis/pushdoanalysis.html.

[18] Shangdong Liu, Jian Gong, Wang Yang, and A. Jakalan. A survey of botnet size measurement. In *Proceedings International Conference on Networking and Distributed Computing*, pages 36–40. IEEE, September 2011.

[19] Luis Corrons. Mariposa botnet, March 2010. http://www.pandasecurity.com/mediacenter/malware/mariposa-botnet/.

[20] Aleksandr Matrosov, Eugene Rodionov, and David Harley. Tdss part 1: The x64 dollar question, 2011. http://resources.infosecinstitute.com/tdss4-part-1/.

[21] McAffee. W32/waledac, February 2009. http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=153670.

[22] McAffee. Mcafee threats report: Second quarter 2012, 2013.

[23] Microsoft. Microsoft, financial services and others join forces to combat massive cybercrime ring, June 2013.

[24] Microsoft. Microsoft security intelligence report volume 14, April 2013.

[25] Microsoft. W32/dorkbot, September 2014. http://www.microsoft.com/security/portal/threat/encyclopedia/Entry.aspx?Name=Win32/Dorkbot#tab=2.

[26] Nart Villeneuve. Koobface: Inside a crimeware network, November 2010. http://www.infowar-monitor.net/reports/iwm-koobface.pdf.

[27] Jose Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 24–31, Oct 2008.

[28] Nicolas Falliere. The sality botnet, May 2010. http://www.symantec.com/connect/blogs/sality-botnet.

[29] Nicolas Falliere and Liam I Murchu and Eric Chien. W32.stuxnet dossier, February 2011.

[30] Peter Mell and Karen Scarfone and Sasha Romanosky. A complete guide to the common vulnerability scoring system version 2.0, June 2007. "http://www.first.org/cvss/cvss-guide.html".

[31] Phillip Porras and Hassan Saidi and Vinod Yegneswaran. An analysis of conficker's logic and rendezvous points, February 2009. http://mtc.sri.com/Conficker/.

[32] Robert J. Bagnall and Geoffrey French. The malware rating system (mrs). "http://www.dodccrp.org/events/6th_ICCRTS/Tracks/Papers/Track7/105_tr7.pdf".

[33] Christian Rossow, Dennis Andriesse, Tillmann Werner, Brett Stone-Gross, Daniel Plohmann, Christian J. Dietrich, and Herbert Bos. P2PWNED: Modeling and Evaluating the Resilience of Peer-to-Peer Botnets . In *Proceedings of the 34th IEEE Symposium on Security and Privacy (S&P)* , San Francisco, CA, May 2013.

[34] Shanmuga. Find and remove spyeye banking trojan, 2011. http://www.malwarehelp.org/find-and-remove-spyeye-trojan-2011.html.

[35] sKyWIper Analysis Team. skywiper (a.k.a. flame a.k.a. flamer): A complex malware for targeted attacks, May 2012.

[36] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Your botnet is my botnet: Analysis of a botnet takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 635–647, New York, NY, USA, 2009. ACM.

[37] Symantec. W32.waledac, December 2008. http://www.symantec.com/security_response/writeup.jsp?docid=2008-122308-1429-99.

[38] Symantec. Internet security threat report volume 18, April 2013.

[39] Symantec. Snapshot of virut botnet after interruption, January 2013. http://www.symantec.com/connect/blogs/snapshot-virut-botnet-after-interruption.

[40] Trend Micro. 3q 2012 security roundup, 2012.