Fight fire with fire: the ultimate active defence

Armando Miraglia and Matteo Casenove
Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Received 15 January 2015
Revised 27 December 2015
Accepted 28 December 2015

Abstract

Purpose – This paper proposes an approach to deal with malware and botnets, which in recent years have become one of the major threats in the cyber world. These malicious pieces of software can cause harm not only to the infected victims but also to actors at a much larger scale. For this reason, defenders, namely, security researchers and analysts, and law enforcement have fought back and contained the spreading infections. However, the fight is fundamentally asymmetric.

Design/methodology/approach – In this paper, the authors argue the need to equip defenders with more powerful active defence tools such as malware and botnets, called antidotes, which must be used as last resort to mitigate malware epidemics. Additionally, the authors argue the validity of this approach by considering the ethical and legal concerns of leveraging sane and compromised hosts to mitigate malware epidemics. Finally, the authors further provide evidence of the possible success of these practices by applying their approach to Hlux, Sality and Zeus malware families.

Findings – Although attackers have neither ethical nor legal constraints, defenders are required to follow much stricter rules and develop significantly more intricate tools. Additionally, attackers have been improving their malware to make them more resilient to takeovers.

Originality/value – By combining existing research, the authors provide an analysis and possible implication of a more intrusive yet effective solution for fighting the spreading of malware.

Keywords Active defence, Botnets, Emerging threats, Malware

Paper type Conceptual paper

Introduction

The threats in the internet are rapidly evolving, becoming more difficult to defeat. Malware writers are improving their code, making their malware and botnets more resilient and robust (Bailey *et al.*, 2009; Rossow, 2013). A bot is software installed on the victim's machine. After installation, the bot joins a network of bots, which is called botnet. The botnet is managed by an operator, which is able to issue commands to the bots for several types of actions. The machine from which the commands are issued is called the command-and-control (C&C). The malicious operations consist in personal-data harvesting, distributed denial-of-service attacks, sending of spam, distributed password cracking, and so on (Holz *et al.*, 2008; Andriesse and Bos, 2013).

Fighting malware is fundamentally asymmetric, favouring the attackers. Although defenders require complicated techniques to mitigate the threats, the attackers require less effort (Sikorski and Honig, 2012). For these reasons, researchers and law enforcement have lately fought back the intrusion by infiltrating and taking over the studied botnets. Examples are Zeus (Ormerod *et al.*, 2010; Andriesse and Bos, 2013), Storm (Holz *et al.*, 2008) and Sality (Falliere, 2011). However, attackers have reacted by improving their botnet architecture to make it more resilient and robust against

Information & Computer Security
Vol. 24 No. 3, 2016
pp. 288-296

2056-4961
DOI 10.1108/ICS-01-2015-0004

infiltrations (Rossow, 2013). The trend suggests that it will become increasingly more difficult to infiltrate botnets by leveraging non-intrusive techniques. Peer-to-peer (P2P) botnets have been shown to be the most resilient among the others. Hence, we focus our study on the P2P botnets, even though the proposed approach is still general.

In this paper, we propose to fill the gap between defenders and attackers by providing the defenders with the very same tools the attackers have at their disposal. More precisely, we believe that for more robust malware infrastructures and more sophisticated attacks, the last resort that the defenders can leverage to defeat the attackers is to use malicious toolkits and active defence techniques. By re-engineering existing malware or developing new ones, defenders will be able to build antidotes to eradicate spreading infections. In this paper, we argue that using these tools, the defender can achieve several goals. First, the antidote can achieve similar host coverage of the malware by exploiting the same spreading techniques. If possible, it can also exploit the vulnerabilities of the malware binary to sanitise the infection. Additionally, when the list of peers is known, the defender can leverage vulnerabilities present in the infected host. As the antidote uses the same protocol to identify new bots, and to spread, it achieves exponential coverage of the infection in the best case. Finally, the antidote is able to monitor on-site the activity of one or several malware, cure the victims and also propagate patches and updates to prevent successive infections. As active defence raises significant ethical and legal concerns, we shed sufficient light on these problems by analysing the key difference between letting the antidote-exploit-infected or sane hosts. Finally, we further argue the effectiveness of the approach by applying it to Hlux, Sality and Zeus P2P malware families.

The remainder of this paper is organized as follows. In the next section, we outline the related work. This is followed by the presentation of our proposed approach, which is then applied to three case study examples of resilient botnets. This then leads into discussion of limitations, and ethical and legal concerns, before finally summarizing our work in the conclusion.

Related work

The related works discussed in this section are the building blocks and the basis of our contributions, setting the ground for the rest of the document.

Rossow (2013) presents and analyses the resilience of botnets. By analysing their overlay network structure and the communication protocol, the work aims to analyse resilience to three types of attacks, enumeration, sinkholing and partitioning. Enumeration consists of gathering information about the topology of the network. Sinkholing is a mechanism aimed to force bots to point to non-existing or fake bot peers. Finally, partitioning aims to split the botnet in disjoint, partially unreachable sub-networks. The work also aims to introduce a scientifically correct research framework to analyse and evaluate the botnet resilience. The analysis has been conducted on several malware families, namely, Storm, Waledac, Zeus, Sality and others. This work was the main inspiring sources for this paper. Based on this research, we have realised the need for different and more intrusive approaches, as the increasing robustness of botnets makes it further difficult to gather information about the botnets and to ultimately disrupt them.

Holz *et al.* (2008) discuss the Storm botnet. The propagation mechanism exploits social engineering. It spreads by means of e-mails that trick the user in installing the malicious

software. Storm used Kademlia structured P2P protocol extended using encrypted messages. The commands distributed to the bots followed a publish-subscribe protocol based on which bots subscribed to keys which the C&C used to publish the commands. Holz *et al.* (2008) present the weakness of protocol, namely, the lack of authentication, used to disrupt the botnet. Holz *et al.* (2008) exploited this vulnerability to conduct enumeration, peer-list poisoning and partitioning. As an impact to our work, the different spreading mechanisms used by this malware type further motivate a thorough analysis of the coverage achievable by emulating the same malicious behaviour. After the disruption of Storm, Waledac emerged. Stock *et al.* (2009) analyse this new malware and identify several commonalities with the Storm botnet. In fact, the propagation mechanism is the same, but the architecture changes, in favour of a more decentralised one. Also, the communication protocols change, hence, the authors took advantage of the malware itself and re-engineered it to infiltrate the botnet and inject servers internet protocol (IP) addresses used for analysis. This paper has an important influence on our work, as the authors re-engineered the malware to infiltrate the botnet. This approach is also considered in our technique, even though we extend this insight and apply a larger scoped methodology, which includes invasive actions. Finally, Bureau (2011) discusses the Hlux (Kehlios) botnet, considered the successor of Waledac. The new malware introduces an additional propagation mechanism, namely, the exploitation of known software vulnerability. The functionalities are roughly the same, but the architecture changes in favour of a hybrid system based on an unstructured P2P protocol. It also makes use of strong encryption routines. Tillmann (2013) presents several attacks on this unstructured P2P botnet. Even though efforts have been put to disrupt this variant, the attackers have been updating and improving the malware binary, making it more difficult to defeat. Hlux is still operational and actively analysed for weaknesses.

Similar to the previous works, Ormerod *et al.* (2010) analyse the first version of the Zeus botnet. Zeus is a malware toolkit, which can be used to easily deploy a botnet. The first version was based on a centralised architecture. On the contrary, Andriesse and Bos (2013) analyse the newer P2P version (Gameover), which is based on an unstructured P2P protocol. Moreover, the newer version improves the previous one by adding strong cryptographic algorithms to protect and authenticate the communication. This new botnet family is still operational. Our analysis takes into consideration the evolution of the Zeus malware. We consider the way in which the malware has been improved to make the protocol more robust and more difficult to infiltrate.

Further, Falliere (2011) analyses the Sality downloader botnet, another P2P resilient botnet. The malware first infects the victim with the downloader, which in turn downloads additional malware components. The botnet relies on peer-local reputation scheme, which is difficult to poison. Moreover, the architecture is extremely resilient to sinkholing. Falliere (2011) analyses several versions of the botnet, but the most significant is Version 4. In fact, this last version corrects two important weaknesses in the previous version, which determine the strong resilience of the botnet, namely, the use of hard encoded repository servers URLs and missing verification of executables to install. As in the case of Zeus, Sality resilience represents an important motivation for our contribution.

Additionally, the importance of early patch application and silent updates spreading is discussed in the context of browsers by Frei *et al.* (2008) Duebendorfer and Frei (2009) and in the context of mobile applications by Tang *et al.* (2012). Moreover, Khouzani *et al.* (2012) further analyse the importance of patching policies to stem the malware

epidemics. These techniques are orthogonal to our approach, as epidemic spreading of the antidote can be accompanied by updates installation. Finally, in the context of local analysis of malware techniques, signature matching (Griffin *et al.*, 2009) or peer traffic analysis (Coskun *et al.*, 2010) can be applied in conjunction with our approach.

Our approach

Based on the evolution outlined above, of some of the major malware observed in the wild, we argue that defenders need to be equipped with more powerful offensive security tools. Figure 1 depicts our approach and the stages needed to set it in place. In this section, we will explain in more detail all the phases and give a better insight about their feasibility.

Background

Before going any further, we discuss two fundamental aspects, which are the basis of our approach. First, this method is not limited to the re-engineering of the targeted malware. In fact, the defender could in principle create a general-purpose antidote. In this case, the developed antidote is able to take advantage of different spreading paths and, hence, mimic various malware-spreading techniques. Moreover, when installed on the victim's machine, the antidote could start the detection phase. If none of the known malware is found, the antidote does not simply react but updates the machine. Differently, if the known malware is detected, the antidote can take further steps to infiltrate or eradicate the infection.

Second, this approach must be the last resort for the defender. In fact, the defender should prefer less invasive approaches. For example, in the case of P2P botnets, if possible, the defender should use sinkholing or partitioning. Differently, when non-intrusive techniques are not available, our approach becomes the only viable option to oppose the infection. The threshold to decide for the presented approach is application-dependent because based on the type of malware targeted, several non-intrusive approaches can be viable.

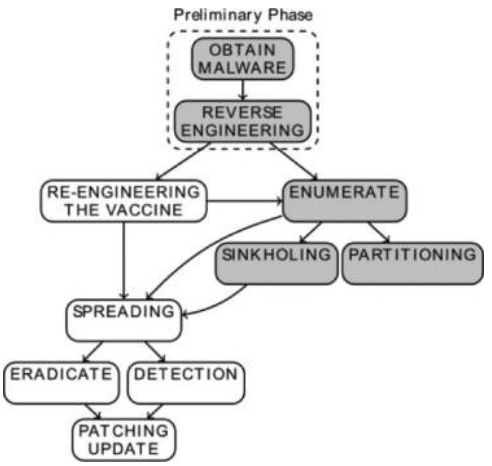

Figure 1.
Flow of actions required to attack a botnet. This figure includes both the currently used approaches and the one described in this paper

Preliminary phase and re-engineering

In the first step, the defender is required to obtain the malware binary. Active or passive methods (Holz *et al.*, 2008) can be used for this purpose. Afterwards, defenders have to apply a thorough reverse engineering activity on the malware sample. This is, in fact, unavoidable, as knowledge of the functionalities of the malware is required. In this way, the defender acquires a deep understanding of the malware protocols and functionalities; and, consequently, they can evaluate the best approaches to emulate and defeat the threat. This initial phase is required by all the approaches targeting malware, no matter how invasive the approach is. The next step consists of re-engineering the malware. Similar to what is described by Stock *et al.* (2009), the defender has to re-engineer the malware sample by disabling the components of the malware which react to commands issued by the C&C, making it harmless, and introducing the new functionalities needed to infiltrate the malware infection, to monitor the malware activities and to exploit weaknesses in the malware or in the victim's software. Hence, our approach allows the defender to exploit the victim's host if necessary. However, as the host is already infected, the exploitation is aimed to help and cure the unknown victim's host, which is executing unwanted and malicious software. At this stage, we call the re-engineered malware antidote. The antidote is a powerful tool, which incorporates many of the properties of the original malware. However, the antidote acts with the goal to spread among the infected hosts to sanitise them and make them immune from future infection attempts.

Spreading

Although the antidote is created, the defender has to take into consideration how the targeted malware spreads. In fact, depending on the technique used, the defender can achieve host coverage similar to the one of the malware. The two main aspects to consider are the exploitation vector and the infiltration mechanism applied.

Exploitation vector. Malware spread quickly as do epidemics for human viruses (Kondakci, 2008), and so they rapidly infect thousands of hosts. Quick spreading is the key objective of botnets, which generally become more resilient when getting bigger. However, from the defender point of view, it is difficult to fight and contain this proliferation. In our approach, we empower the defender with the same spreading tools to improve the effectiveness of their actions. It is important to notice, however, that the defender must always make sure that the targeted victim is, in fact, a threat to avoid further non-required actions. The techniques are described below:

- *Use the same exploitation vector used by the malware*: Some of the malware analysed in the related work section spread by exploiting known vulnerabilities. This is the case for Hlux. The analyses reviewed do not mention any action taken by the malware to patch the vulnerability exploited, and, hence, we can safely assume that such a step is not taken. Consequently, the defender can produce an antidote that uses the same exploitation vector used by the malware. This will guarantee that the antidote spreads in a similar way the targeted malware does, achieving similar host coverage. Hence, by blindly exploiting the same vulnerabilities, the antidote has a high probability to hit a host that is infected. Finally, this approach provides also the ability for the defender to take advantage of zero-day vulnerabilities that the malware also exploits.
- *Take advantage of vulnerability in the malware software*: If the previous approach is not possible because the malware spreads using social engineering techniques, such

as Sality and Zeus, or patches of the used vulnerabilities, a different approach can be used. In fact, even though simpler than the defenders' tools, malware are still intricate software, which can be affected by software vulnerabilities. In this case, the antidote can exploit malware vulnerabilities to spread on the infected machines and sanitise them. In this case, the victim's host is guaranteed to be infected, because otherwise, the exploitation would not be possible.

- *Take advantage of vulnerability in the victim's host*: Moreover, if the host is known to be infected and the options previously presented are not viable, the defender can exploit the victim's host. In this case, known vulnerabilities or zero-day vulnerabilities known by the defender can be used to put the antidote on the infected host and further work on it.
- *Use black-market services*: Finally, considering that many attackers take advantage of specialized tools such as downloaders (Rossow, 2013), the defender can act undercover and buy the provided service. This is an additional spreading vector to achieve similar malware coverage.
- *Enumeration and sinkholing*: Additionally, Figure 1 depicts another important aspect of spreading. In case the defender was successful in enumerating the malware instances, it could directly address them with the antidote, and a similar action could be performed if the targeted botnet has been sinkholed. The antidote can be directed to the known infected hosts.

Infiltration. If a victim is detected or the antidote was started on a honeypot already running the malware, the antidote can attempt to infiltrate the malware. The antidote can try to overtake the process and obtain information related to the malware. In case of a botnet, it can try to obtain the peer list stored in memory. In other cases, it can observe and takeover the malware by observing and fixing infected files. The main difference of the infiltration phase of our approach with respect to the currently used techniques is that our approach allows the defender to overtake a running malware binary on the victim's host and mimic its behaviour.

Detection and eradication. When the antidote is on-site, it can take additional actions. A general-purpose antidote can leverage detection techniques (Griffin *et al.*, 2009; Coskun *et al.*, 2010) to verify if a known malware family for which the antidote is programmed has already infected the host on which the antidote is installed. Differently, an antidote is intended to be deployed on hosts that are known to be infected by the targeted malware. In this case, the antidote is deployed after an initial information gathering about the presence of the malware on the host. However, it is still possible that the malware is not operational on the host. Therefore, as previously mentioned, it is important that the antidote further checks for the threat. After the antidote is able to determine the ongoing and active infection and collects the information required, it can eradicate the malware. This can be more or less difficult depending on type of malware targeted.

Patching and update. Finally, the antidote can fix known bugs and exploited vulnerabilities by forcing updates and patches application on the victim's machine. As suggested by Duebendorfer and Frei (2009), Khouzani *et al.* (2012) and Tang *et al.* (2012), the importance of updates application is the key point to stop an epidemic. Moreover, silent updates are further more important, as they can achieve much better protection and prevention. This is a key side effect that the antidote can leverage. In this manner, the

antidote improves the resilience of the victim's host against new attacks. Social engineering attacks are yet very difficult to sanitise. Hence, patching and updating are not sufficient against these types of spreading techniques. However, the antidote can use additional countermeasures such as blacklisting of known bad domains and e-mail addresses.

Case study

To understand how this approach can be effective against epidemic spreading and malware countermeasure, we briefly present applications of our approach to real-world malwares. We envision the result of our approach by discussing the steps that can be followed to take down three resilient botnets.

Sality and Zeus P2P

The Zeus P2P malware leverages the drive-by-download – by means of browser vulnerabilities or other security flaws, and the malware is automatically installed when the user "drives by" the infectious website – technique to infect new machines, whereas Sality uses social engineering to spread. In both cases, the antidote cannot take advantage of the malware-spreading technique easily, but it can start operating from an infected honeypot. From this point, the antidote can both takeover the local running malware and steal useful information, similar to the peer list. Moreover, by targeting the infected hosts, the antidote spreads and applies the cure to the other hosts. In the case of Sality, the antidote can mimic the infectious behaviour and sanitise as many binary files as possible. By using the same malware procedure, there is high probability to cover many of the binary files infected by the malware. Finally, the antidote can disrupt the malware functionalities, for example, by sniffing and dropping the communication. To keep the protocol from further spreading, the antidote can emulate the behaviour of the on-site malware to remain in the malicious network and further acquire information about the infected hosts.

Hlux

Different from what was presented above, the latest versions of Hlux exploit a well-known Windows vulnerability issue (Bureau, 2011) to achieve better epidemic. As suggested earlier in this paper, by preparing an antidote that uses this vulnerability, it has more chances to hit an already infected host. We are again assuming that the malware does not fix the issue after installing itself on the victim.

Discussion

In this section, we consider limitations and press on ethical and legal controversial issues related to the presented technique.

Limitations

First, our approach still requires a lot of reverse engineering, which is unavoidable unless, as in the case of Zeus, the source code is publicly available. The techniques are heavily dependent on the family of malware targeted. In fact, especially for infiltration, it is important that the antidote is able to fully imitate a bot/peer of the malware family targeted. Moreover, to obtain similar coverage, the exploitation of the same proliferation mechanisms is required. Even though techniques exist that are able to detect a malware, polymorphism and encryption are still difficult to defeat. Even for instrumenting the malware with detection mechanisms, currently available techniques are still not powerful enough to always detect the presence of malware with 100 per cent accuracy.

Ethical and legal concerns

The approach presented in this paper raises ethical and legislative concerns. On the one hand, by exploiting and invading the bot machine, the defender would, in fact, invade the victim's privacy. The victim would be considered responsible for "negligence in use of electronic and telecommunication devices", which is illegal in most democracies. Recently, several cases have been disclosed in which offensive security has been used for espionage and communication disruption (Jaeger, 2012; Greenwald, 2013; Poulsen, 2013). In most of the countries afflicted by these actions, the reaction has been to punish the authors of the actions. Moreover, some governments have also introduced *ad hoc* laws (Jaeger, 2012) to discourage the use of such methods. In all these happenings, the public opinion has always arisen against the surveillance activities. On the other hand, the actions punished clearly aimed to monitor and limit the freedom of speech. On the contrary, actions required by the proposed approach are directed to protect the common good and defeat criminals' intents. By carefully designing a legislative framework in which offensive actions can be considered legal when fighting criminals and preserving the citizens' freedom, the approach proposed could, in fact, become a powerful tool to mitigate the threat represented by malware and botnets. Moreover, if taking over a sane host is questionable, we believe that attacking an infected host has a fundamentally different meaning, as the host has been already compromised and is performing illegal actions. Based on these observations, it is hard to believe that active defence will be considered a viable option to fight botnets and malware. However, the authors believe that in the future, the need to fight criminals will prevail.

Conclusions

In this paper, we motivated the use of active defence by the defenders to mitigate malware epidemics. Opposing malicious software spreading and fighting criminals in the cyber world are asymmetric challenges. Attackers can easily develop simple and compact codes to obtain their illegal intents. On the contrary, defenders have to develop significantly more intricate and complex tools to oppose and track down these attackers. Moreover, attackers are improving their malicious software and architectures to be significantly more resilient to take down attempts. For this reasons, in the future, defenders will need active defence and offensive security tools to fight back malware and infections. We presented a structured technique to turn a malware binary into an antidote, which leverages the malware characteristics to counterattack the infection. We further described the application of our technique to mitigate Sality and Zeus P2P, two resilient botnets. Finally, we proposed a discussion about the technical limitations of our approach and the ethical and legal concerns that arise from the presented approach.

References

Andriesse, D. and Bos, H. (2013), "An analysis of the Zeus peer-to-peer protocol", Technical Report, Vrije Universiteit Amsterdam, Amsterdam.

Bailey, M., Cooke, E., Jahanian, F., Xu, Y. and Karir, M. (2009), "A survey of botnet technology and defenses", *CATCH'09*, IEEE Computer Society, Washington, DC, pp. 299-304.

Bureau, P. (2011), "Same botnet, same guyes, new code: Wn32/Kelihos", Technical Report, VirusBulletin, Abingdon OX.

Coskun, B., Dietrich, S. and Memon, N. (2010), "Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts", *ACSAC'10*, ACM, New York, NY, pp. 131-140.

Duebendorfer, T. and Frei, S. (2009), "Why silent updates boost security", Technical Report, TIK, ETH Zurich, available at: www.techzoom.net/silent-updates

Falliere, N. (2011), "Sality: story of a peer-to-peer viral network", Technical Report, Symantec Corporation, Mountain View, CA.

Frei, S., Duebendorfer, T. and Plattner, B. (2008), "Firefox (In) security update dynamics exposed", *SIGCOMM Computer Communication Review*, Vol. 39 No. 1, pp. 16-22.

Greenwald, G. (2013), "NSA collecting phone records of millions of Verizon customers daily", *The Guardian*, available at: www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order

Griffin, K., Schneider, S., Hu, X. and Chiueh, T.C. (2009), "Automatic generation of string signatures for malware detection", *RAID'09*, Springer-Verlag, Berlin, Heidelberg, pp. 101-120.

Holz, T., Steiner, M., Dahl, F., Biersack, E. and Freiling, F. (2008), "Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm", Usenix LEET'08, pp. 1-9.

Jaeger, M. (2012), "Germany backs away from using a Trojan on its citizens – for now", available at: www.zdnet.com/germany-backs-away-from-using-a-trojan-on-its-citizens-for-now-7000008914/

Khouzani, M., Eshghi, S., Sarkar, S. and Venkatesh, S.S. (2012), "Optimal patching in clustered malware epidemics", *INFOCOM*, San Diego, CA, pp 1-9.

Kondakci, S. (2008), "Epidemic state analysis of computers under malware attacks", *Simulation Modelling Practice and Theory*, Vol. 16 No. 5, pp. 571-584.

Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M. and Wang, L. (2010), "On the analysis of the Zeus botnet crimeware toolkit", *PST'10*, Ottawa, ON, pp. 31-38.

Poulsen, K. (2013), "FBI admits it controlled tor servers behind mass Malware attack", available at: www.wired.com/threatlevel/2013/09/freedom-hosting-fbi/ (accessed 13 September 2013).

Rossow, C. (2013), "Using malware analysis to evaluate botnet resilience", PhD thesis, VU Amsterdam.

Sikorski, M. and Honig, A. (2012), *Practical Malware Analysis*, No Strach Press, San Francisco, CA.

Stock, B., Goebel, J., Engelberth, M., Freiling, F.C. and Holz, T. (2009), "Walowdac – analysis of a peer-to-peer Botnet", *EC2ND '09*, IEEE, Washington, DC.

Tang, J., Kim, H., Mascolo, C. and Musolesi, M. (2012), "STOP: socio-temporal opportunistic patching of short range mobile malware", *WOWMOM*, IEEE Computer Society, San Francisco, CA, pp. 1-9.

Tillmann, W. (2013), "Peer-to-peer poisoning attack against the Kelihos.C Botnet", available at: www.crowdstrike.com/blog/peer-peer-poisoning-attack-against-kelihosc-botnet/

Further reading

Doucear, J.R. (2002), "International workshop on peer-to-peer systems", *IPTPS '01*, Springer-Verlag, London, pp. 53-65.

Corresponding author

Armando Miraglia can be contacted at: a.miraglia@student.vu.nl