Research Paper:

Applying epidemiological models of interacting epidemics to the context of computer malware

Letitia Kernschmidt, B.Sc.

Vienna University of Economics and Business

Contents

List of abbreviations

ADE	Antibody-Dependent Enhancement
HIV	Human Immunodeficiency Virus
WWW	World Wide Web

1 Introduction

> I think computer viruses should count as life. I think it says something about human nature that the only form of life we have created so far is purely destructive. We've created life in our own image.
>
> *Stephen Hawking, Macworld Expo, Boston 1994*

In 2016, security researchers discovered approximately 357 million new unique malware variants, which posed a significant risk to governments, corporations, and private persons [Cor16]. Accordingly, governments and companies alike require reliable models that can be used to analyze and forecast the spreading dynamics of malware in order to develop adequate prevention, detection and immunization strategies. Therefore, a thorough understanding of the different spreading dynamics of malware is required. The findings from research on epidemic spreading of infectious diseases have thereby built the basis for research on the spreading dynamics of computer malware (see, e.g. [KCW93]).

In the context of epidemic spreading of infectious diseases especially the spreading dynamics of the most dangerous, i.e. lethal, infectious diseases such as the Human Immunodeficiency Virus (HIV), tuberculosis and malaria [Org15], have been extensively studied and respective models have been introduced. With the emergence of the internet and the World Wide Web (WWW) and the rising number of malware spreading across computer networks, researchers increasingly transferred these epidemiological models of infectious diseases to the context of computer malware.

Research on infectious diseases, however, has shown that there are often interactions between two or more pathogens that spread across the same network and compete for the same hosts. Interacting behavior can be repeatedly also observed between two different malicious software programs. For example, once a computer is infected with the so-called *Shifu* banking Trojan (first detected in October 2015), the malware monitors all subsequent downloads and deletes other malicious programs [Cor15]. In this way, *Shifu* can ensure to exploit the target exclusively. At the same time, this means that the victim imparts in a way cross-immunity against other malware. On the other hand, *Shifu* can be also used to download more malware from the internet thereby acting as an amplifier of the victim's contagiousness [Cor15]. In the context of computer malware, however, there exist also special types of interactions that are not known from infectious diseases (or only in similar forms) like for example so-called *turf wars*. In this case, a malicious software program is intentionally designed to exploit or destroy another specific malware (see, e.g., [Her07]). For example, until December 2009, *Zeus* was the most established toolkit for setting-up botnets in order to steal online banking credentials [McM10]. At the beginning of 2010, however, researchers detected a feature called *Kill Zeus* in the *Spy Eye* toolkit, which was a much smaller rival of *Zeus* [Coo10]. The *Kill Zeus* feature sought for computers already infected with a *Zeus* Trojan, then stole the data and removed the rivaling malicious software program from the infected computer [McM10].

1.1 Objective

As described above there are various ways in which two pathogens spreading across the same network and competing for the same hosts can interact with each other. Therefore, one aim of this thesis is to give an overview of these different types of interactions. The main goal, however, is to apply an existing model of interacting epidemics from the biological field to the context of computer malware. In particular, this thesis focuses on the case of increased contagiousness towards a specific disease (or malware, respectively) due to prior infection with a different disease or malware, respectively. Using computer simulations, outbreaks of two different malicious software programs, which exhibit interacting behavior, are reproduced. Thereby, different graph topologies are taken into account in order to examine the influence of the graph structure on the spreading dynamics. In particular, the simulations are run on eight different graphs, from which two are real-life graphs (a smaller one with 1,133 vertices and 5,451 edges and a larger one with 6,474 vertices and 12,572 edges) and six are generated graphs (two $G(n,p)$ random graphs, two scale-free graphs, and two small-world graphs each of the sizes of the real-life graphs). Furthermore, the influence of the starting point of the epidemic outbreak is considered. Therefore, the number of initially infected vertices as well as the selection criteria of the initially infected vertex/vertices are modified. In particular, the number of initially infected vertices is varied between one infected vertex and one percent of infected vertices. This/these initially infected vertex/vertices is/are selected either randomly, based on the Betweenness centrality or based on the Eigenvector centrality.

1.2 Structure

The remainder of this thesis is structured as follows. Chapter 2 introduces relevant terminology and concepts of networks and graph theory. Chapter 3 gives an overview of the history of epidemics as well as important definitions and distinctions and relevant models. A classification of malware is given in Chapter 4. Interacting epidemics are introduced in Chapter 5. Then, Chapter 6 presents the design and implementation of the simulations as well as the results and the respective discussion. Finally, Chapter 7 concludes the thesis.

2 Networks and Graphs

Although the terms *networks* and *graphs* are mostly used synonymously in the scientific literature, there is a subtle difference between these two terminologies. Whereas the term *network* mostly refers to some real-world system such as the WWW, protein networks or social networks, the term *graph* is used when the mathematical representation of these networks is discussed. Throughout this thesis, the terms will be used according to this differentiation.

Note that, if not stated otherwise, the following explanations are all based on [vS10], [New12], and [Bar14].

2.1 Basic terminology

A *graph* consists of *vertices* and *edges*. The *order* of a graph, denoted by $|V(G)|$, $|V|$ or via the variable n, is the number of all vertices and the *size* of a graph, denoted by $|E(G)|$, $|E|$ or via the variable m, the number of all edges. In general, each edge connects (or *joins*) exactly two vertices, which are then said to be *adjacent* or *neighbors*. The connecting edge, on the other hand, is said to be *incident* with the two vertices.

Definition 1: Graphs

A **graph G**, denoted by $G = (V, E)$, consists of a set of **vertices** $\boldsymbol{V}$ and a set of **edges** $\boldsymbol{E}$[a]. Each edge $e \in E$ **joins** one or two vertices, which are then said to be the edge's **end points**. An edge $e \in E$ that joins $u, v \in V$, is denoted by $e = \langle u, v \rangle$.

[a] A graph consisting solely of vertices without any edges is called an *edgeless graph*. Besides this rather rare instance, however, all other types of graphs consist of both vertices and edges.

2.1.1 Undirected Graphs

In case of *undirected graphs* the edges are represented as unordered pairs of vertices. Consequently, there is no difference between the edges $\langle u, v \rangle$ and $\langle v, u \rangle$. Note that throughout this thesis only *undirected graphs* are considered.

2.1.2 Subgraphs

Subgraphs are used to focus on those subsets of vertices and edges, which are relevant for solving a certain problem. Therefore, all irrelevant vertices together with the associated edges are temporarily ignored in order to obtain a certain subgraph.

Definition 2: Subgraphs

A graph *H*, denoted by *H* = *(V,E)*, is a **subgraph** of graph *G*, if *V(H)* ⊆ *V(G)* and *E(H)* ⊆ *E(G)*. The relationship between graph *G* and its associated subgraph *H* is denoted by *H* ⊆ *G*.

2.1.3 Degree

The number of all edges incident to a vertex is called the *degree* of the vertex.

Definition 3: Degree

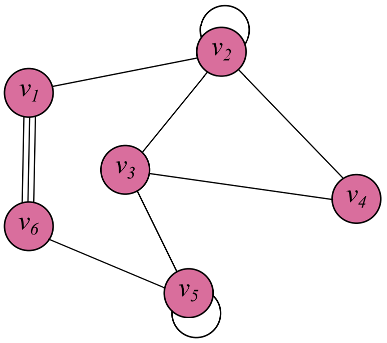
In an undirected graph, the **degree** of a vertex *v* ∈ *V(G)*, denoted by ***δ(v)***, is the number of edges incident with *v*.

2.2 Graph representations

Although there are multiple possible graph representations, only the adjacency matrix and the edge list representation are used throughout this thesis and are hence presented in more detail in the following sections.

2.2.1 Adjacency matrix

The *adjacency matrix* of a graph *G*, denoted by A_G, is an *n* × *n* matrix with each entry a_{ij} indicating the number of edges between two vertices v_i and v_j. If there is no edge between the two vertices, the entry takes the value 0. Figure 1 shows an example for an undirected graph G.



$$A_G = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 3 \\ 1 & 2 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 1 \\ 3 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 1: **A graph *G* with its associated adjacency matrix A_G**

Most graphs that represent a real-world network are sparse. This means that they have a significantly smaller number of edges than a *complete graph* with the respective number of vertices. A complete graph is a graph, in which every vertex is connected to every other vertex in the graph. Accordingly, the associated adjacency matrices of graphs representing real-world networks are also sparse. As a consequence, especially in the case of large graphs, it proves to be more efficient to store only the list of actually existing edges instead of the whole adjacency matrix containing all the entries equal to zero.

2.2.2 Edge list

Due to the more efficient way of storage, *edge lists* may be the preferred choice of representation when dealing with large sparse graphs. The edge list for the graph presented in Figure 1 is as follows:

$$(\langle v_1, v_2\rangle, \langle v_2, v_2\rangle, \langle v_2, v_3\rangle, \langle v_2, v_4\rangle, \langle v_3, v_4\rangle, \langle v_3, v_5\rangle, \langle v_5, v_5\rangle, \langle v_5, v_6\rangle, \langle v_6, v_1\rangle, \langle v_6, v_1\rangle, \langle v_6, v_1\rangle)$$

Whereas $n \times n$ elements need to be stored for any adjacency matrix, the edge list grows only linearly with the number of edges.

2.3 Degree distribution

Recall that in an undirected graph G, the degree δ of a vertex v is the number of edges attached to v (see Section 2.1.3). Hence, the degree distribution of a graph G is defined as follows.

Definition 4: Degree Distribution

The fraction of vertices having degree δ is computed as follows:

$$frac_\delta = \frac{h(\delta)}{n} \tag{2.1}$$

where $h(\delta)$ is defined as the number of vertices having degree δ. The quantities of $frac_\delta$ represent the *degree distribution* of a graph G.

2.4 Distances

A central problem in connected graphs is finding the shortest path between two randomly chosen distinct vertices u and v. The length of the *shortest path* P_{short} (also called *geodesic*) between two vertices u and v is the *(geodesic) distance* $d(u, v)$. In case of an undirected unweighted graph G, the shortest path is the path with the smallest edge count. There can exist none, one or several shortest paths between two distinct vertices.

Definition 5: Distance metrics

The *eccentricity* of a vertex v in a connected graph G is defined as the maximum distance between v and any other vertex $u \in V(G)$, i.e. $ecc(v) = max\{d(v,u)|u \in V(G)\}$. The minimum over all eccentricity values of a connected graph is called *radius*, i.e. $rad(G) = min\{ecc(v)|v \in V(G)\}$. In contrast, the *diameter* of a connected graph G is defined as the maximum over all eccentricity values of G, i.e. $diam(G) = max\{ecc(v)|v \in V(G)\}$.

Moreover, several metrics for analyzing the distribution of path lengths exist.

Definition 6: Path lengths

In a connected graph G the **average length of the shortest paths** from a vertex v to any other vertex u is computed as follows:

$$\bar{d}(v) = \frac{1}{n-1} \sum_{v \in V, u \neq v} d(v,u) \tag{2.2}$$

Accordingly, the **average path length of *G*** is defined as

$$\bar{d}(G) = \frac{1}{n} \sum_{v \in V} \bar{d}(v) = \frac{1}{n^2 - n} \sum_{u,v \in V, u \neq v} d(v,u) \tag{2.3}$$

2.5 Clustering coefficient

The clustering coefficient (also referred to as *graph transitivity* $\tau(G)$) reveals to what extent the neighbors of a vertex v are neighbors themselves.

Definition 7: Graph transitivity (Global clustering coefficient)

The **graph transitivity $\boldsymbol{\tau(G)}$** is defined as the ratio between the number of **triangles** and the number of **triples** in a graph G. A triangle at a vertex v is a complete subgraph $H \subseteq G$ with exactly three vertices, including v. A triple at a vertex v is a subgraph $H \subseteq G$ with exactly three vertices and two edges, where v is incident with both edges. The graph transitivity is hence defined as

$$\tau(G) = \frac{n_\Delta \times 3}{n_\Lambda} \tag{2.4}$$

where n_Δ is the number of triangles and n_Λ the number of triples in a graph G.

2.6 Centrality

Centrality deals with the question whether there are vertices in a graph that are “more important” than other ones. Within this thesis only the Eigenvector centrality and the Betweenness centrality are considered in more detail.

2.6.1 Eigenvector centrality

The *Eigenvector centrality* is based on the knowledge that a person (i.e. a vertex) may not only be important if he or she knows a lot of other people (i.e. has a high degree) but also if the person knows only a few yet very important individuals. In order to consider this relative importance of a vertex with respect to its neighbors, Phillip Bonacich introduced the Eigenvector centrality [Bon72]. In particular, the Eigenvector centrality of a vertex v is proportional to the sum of the Eigenvector centralities of its neighbors, i.e.

$$C_{Eig}(v) = \frac{1}{\lambda_{dom}} \sum_{u} a_{vu} C_{Eig}(u) \tag{2.5}$$

where λ_{dom} is the dominant (largest) eigenvalue of the adjacency matrix of a graph G.

2.6.2 Betweenness centrality

Another concept of centrality is called *Betweenness centrality*. In this case the importance of a vertex v is dependent on the number of shortest paths between two vertices that go through v. This centrality measure considers such a vertex v important, because the removal of v might directly influence the connectivity between two other vertices in a graph. This means that another vertex might not be reachable anymore after the removal of the vertex v.

For a connected graph G, $S(u, w)$ is the set of shortest paths between two vertices $u, w \in V(G)$ and $S(u, v, w) \subseteq S(u, w)$ the set of shortest paths between u and w that go through $v \in V(G)$. The Betweenness centrality is then defined as

$$C_{Bet}(v) = \sum_{u \neq v \neq w \in V(G)} \frac{|S(u, v, w)|}{|S(u, w)|} \tag{2.6}$$

2.7 Random graphs

In general, a *random graph* is an undirected, simple and connected graph G in which certain parameters take fixed values while the other properties of the graph are random.

Definition 8: The G(n,p) Model

In case of the $\boldsymbol{G(n,p)}$ **Model**, the number of vertices n and the probability p that two vertices are connected by an edge take fixed values. Accordingly, two $G(n, p)$ random graphs with the same fixed values for n and p might vary widely.

2.8 Small worlds

The *small world phenomenon* was first studied by Stanley Milgram in 1967 [Mil67]. In particular, Milgram's research focused on social networks and the question of how far (i.e. by how many acquaintances) any two individuals in Kansas, Nebraska and Massachusetts are separated from each other. Therefore, Milgram conducted an experiment in which he sent hundreds of letters to people in the Mid-West of the U.S. including further instructions. If the recipient of the letter knew the target in person, he/she should directly forward the letter to the target living in Massachusetts. If the recipient, however, did not know the target in person, he/she should forward the letter to another person of whom he/she thought could know the target. On average, it took only 5,5 hops until a letter reached the target.

In general, the outcome of this and numerous subsequent experiments was that the average path length in small worlds is relatively small. In this aspect, small worlds conform with $G(n, p)$ random graphs (see Section 2.7). They differ, however, with respect to the clustering coefficient. Whereas $G(n, p)$ random graphs have a clustering coefficient equal to p, small worlds appear to have a very high clustering coefficient. This is, because in social networks the acquaintances of one person tend to be acquaintances themselves.

2.9 Scale-free graphs

When analyzing numerous real-world networks, researchers figured out that the degree distribution often does not follow a Poisson distribution as it is the case for $G(n, p)$ random graphs, but rather a *power-law*.

In random graphs all vertices have approximately the same degree and hence the average degree μ_δ acts as the scale of a graph. In contrast, the degree of a randomly chosen vertex cannot be approximated beforehand if the degree distribution follows a power law. Accordingly, these graphs lack a scale and are thus called *scale-free graphs*. Scale-free graphs are characterized by a few vertices with a very high degree (called *hubs*) and a large number of vertices with a very low one.

3 Epidemics

The term *epidemic* originates from the Greek preposition *epi* (ἐπί) meaning *upon* and the noun *demos* (δῆμος) meaning both *people* and *homeland* [Gem97]. The original meaning of *epidemos* (ἐπί-δήμος) was hence *in the community* or *indigenous* [Gem97] and was not used in a medical sense. This changed in the 5th century BC, when the famous ancient doctor Hippocrates used this term in his works *Of the Epidemics*. In particular, he was probably the first to use this expression in order to describe diseases that *circulate or propagate in a country* [MMG06]. Although his works focused more on the relationships between diseases and environmental factors, such as the habitat of the patients or the seasons in which the diseases occurred, than on the spreading behavior, he nonetheless significantly coined the term epidemics in its now known medical sense [MMG06].

Throughout the following centuries, countless authors documented various epidemic outbreaks. However, they almost exclusively focused rather on the symptoms of the disease than on the spreading dynamics or the source of the disease, with Daniel Bernoulli's epidemiological paper (published in 1766) being a rare exception (see, e.g. [Bai75]). In particular, Bernoulli presented a mathematical model in order to calculate the gain in life expectancy at birth if smallpox were eradicated. Hence, this paper is often assumed to be the first one to deal with population dynamics of infectious diseases (see, e.g. [Bai75]). It was, however, not until the 19th century that major progress was made in this field [Bai75].

A story very often referred to as the origin of modern epidemiology is the story of Dr. John Snow, who allegedly stopped a severe cholera epidemic in London in 1854 after identifying the Broad Street water pump as the source of this cholera outbreak. It is told that he found the correlation between the contaminated water pump and the spreading of the cholera disease by plotting the cholera deaths on a street map. Although there is hardly any preserved evidence for Snow's actual contribution to the end of the outbreak, his notes can still be seen as one of the first documents of modern epidemiology [McL00].

Twenty-two years later, the scientific field of epidemiology received broad public attention after Robert Koch's discovery of the bacteria *Bacillus Anthracis* as the vector of anthrax. During this time, the Russian physician Pyotr Dimitrievich En'ko published a probabilistic model and thorough data analysis of measles epidemics in the late 1880s (see, e.g., [CFG$^+$01]). Several authors refer to this paper as the first contribution to modern mathematical epidemiology (see, e.g., [CFG$^+$01]).

At the beginning of the 20th century more sophisticated epidemiological models were introduced and since the middle of the 20th century the scientific field of mathematical epidemiology has grown rapidly resulting in the introduction of a great variety of generic and specialized deterministic and stochastic models (for an overview see, e.g., [Bai75]).

3.1 Epidemiological Models

3.1.1 Basic Assumptions

The model discussed in this thesis is a so-called *compartmental model*. This means that the population under consideration is divided into several disjoint classes[1] that change with time t [Het89]. For the compartmental epidemiological model presented in Section 3.1.2, the following assumptions hold true (see, e.g., [Het89]):

- The *population size n is constant*.
- The *population is homogeneously mixing*. This means that each individual in the population has the same chance of contacting every other individual in the population.
- The *latent period is zero*. This means that a person who is exposed to the disease is immediately infected.

3.1.2 The epidemic SIR-Model

The SIR-Model (Susceptible-Infected-Recovered Model) is used to model infections from which infected individuals recover with immunity. It consists of the three classes *susceptible*, *infected* and *recovered* (see, e.g., [New12]). The respective flow chart of the SIR-Model is depicted in Figure 2.

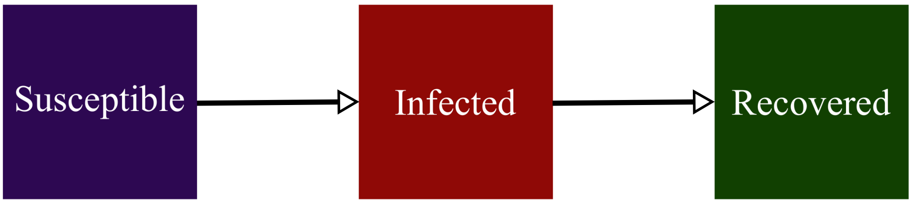


Figure 2: Epidemic SIR-Model

Individuals in the susceptible class are not yet infected, but might be infected when getting into contact with an individual from the infected class. Individuals in the infected class are infected with the disease and can eventually recover from the disease. Individuals in the recovered class have recovered from the disease with immunity and cannot transmit the disease to any other individual. Infected individuals are assumed to recover (or die) from the disease at a constant average *recovery rate* γ. Hence, the *average duration of infection* is equal to $1/\gamma$.

Considering the fractions of the classes (with $s + i + r = 1$), the ordinary differential equations for the rates of change of the three classes, can be formulated as follows (see, e.g., [New12]):

[1] In the case of the basic epidemiological models the population under consideration is divided into three disjoint classes.

$$\frac{ds(t)}{dt} = -\beta s(t)i(t) \tag{3.1}$$

$$\frac{di(t)}{dt} = \beta s(t)i(t) - \gamma i(t) \tag{3.2}$$

$$\frac{dr(t)}{dt} = \gamma i(t) \tag{3.3}$$

Eliminating $i(t)$ between Equations 3.1 and 3.3 results in [New12]

$$\frac{1}{s(t)}\frac{ds(t)}{dt} = -\frac{\beta}{\gamma}\frac{dr(t)}{dt} \tag{3.4}$$

which can be integrated on both sides with respect to t:

$$s(t) = s_0 e^{-\beta r(t)/\gamma} \tag{3.5}$$

where s_0 is the value of s at time $t = 0$ and where the number of recovered individuals at time $t = 0$ equals zero. Using $i(t) = 1 - s(t) - r(t)$ and feeding Equation 3.5 in Equation 3.3 results in:

$$\frac{dr(t)}{dt} = \gamma(1 - r(t) - s_0 e^{-\beta r(t)/\gamma}) \tag{3.6}$$

Based on Equation 3.6, the *total size* of the epidemic, i.e. the total number of all individuals, who get infected with the disease during the entire time of the epidemic outbreak, can be calculated as the value at which $dr(t)/dt = 0$ (for more details see, e.g., [New12]). Assuming that in the limit of a large population size $n \to \infty$ the initial value $s_0 \approx 1$, the final value of $r(t)$ is determined by [New12]

$$r(t) = 1 - e^{-\beta r(t)/\gamma} \tag{3.7}$$

3.1.3 Degree Block Approximation

As described above, a basic assumption of the deterministic epidemic models presented above is that the population under consideration is homogeneously mixing, i.e. that each individual in the population has the same chance of contacting every other individual in the population. This assumption is dropped in case of *degree block approximation*, in which case the probability of infection of a vertex is dependent on its degree (see, e.g., [Bar14]). This means that a vertex with a higher degree is more likely to be in contact with an infected vertex than a vertex with a smaller degree[2].

Similar to the compartments described above (i.e. *susceptible*, *infected*, and *recovered*), an additional set of compartments is introduced in case of degree block approximation. In particular,

[2] Note that there are also several other mathematical models for epidemic spreading in contact networks (e.g. *pairwise models* or *effective degree models*). However, since degree block approximation was implemented for the simulations in this thesis (i.e. vertices with the same degree are treated similarly) only this model is described in more detail. For a comprehensive overview of existing models see, e.g., [MK14].

all vertices with the same degree are placed into the same compartment (i.e. block) assuming hence that all vertices with the same degree behave equivalently (see, e.g., [Bar14]). Note that the blocks for each degree are additionally added to the existing compartments of the epidemic model. This means that each vertex can be in a susceptible, infected, or recovered (in case of the SIR model) state independent of its degree. A visualization of degree block approximation for an example graph G and the SIR epidemic model is given in Figure 3.

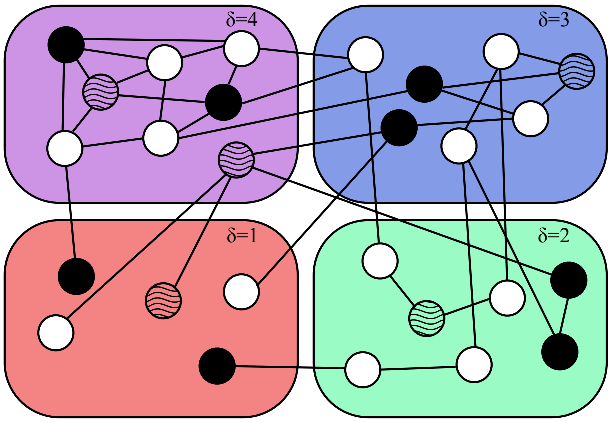


Figure 3: Degree Block Approximation for an example graph G with degrees $\delta = 1$, $\delta = 2$, $\delta = 3$, and $\delta = 4$. The vertices are either in the susceptible (white), infected (black), or recovered (wave pattern) state. Figure based on [Bar14]

The fraction of infected vertices with degree δ_k among all vertices with degree δ_k is denoted by

$$i(t)_\delta = \frac{I_\delta}{n_\delta} \tag{3.8}$$

Hence, the differential equation of the SI-model can be separately written for each degree [Bar14]:

$$\frac{di(t)_\delta}{dt} = \beta(1 - i(t)_\delta)\delta\theta_\delta \tag{3.9}$$

with the density function θ_δ representing the fraction of infected neighbors of a susceptible vertex with degree δ.

Similarly, the differential equation for the SIR-model (see Equation 3.2) can be separately rewritten for each degree as [Bar14]

$$\frac{di(t)_\delta}{dt} = \beta s_\delta \theta_\delta - \gamma i(t)_\delta \tag{3.10}$$

with $s_\delta = 1 - i_\delta - r_\delta$.

4 Malware

The term *malware* (a composition of the two terms *malicious software*) is an umbrella term for all programs that pose a significant security risk to a user's system or (personal) information by performing unauthorized and/or harmful actions (see, e.g., [OEC09]). Attackers use malware to steal personal or program data, to secretly manipulate the computer or the installed programs, to completely block the user from using the device or otherwise affect the data and system integrity (see, e.g., [OEC09]). For the purposes of this thesis, different types of malware are categorized depending on the type of transmission. Table 1 gives an overview of the different types of malware as defined for the purpose of this thesis.

Malware Type	**Definition**
Computer Virus	A *computer virus* is defined as "a program that recursively and explicitly copies a possibly evolved version of itself" [Szo05] when executed. All viruses can only infect other programs on the same computer, but are not able to spread autonomously over a network. Therefore, they are mostly transmitted via downloads from the internet, removable physical devices, and *phishing* e-mail attachments. A phishing e-mail is an e-mail, with a malicious attachment or link that is sent to a victim with the intent of tricking the recipient to open the attachment or click on the link.
Computer Worm	A *computer worm* is defined as a self-contained, self-propagating and self-replicating computer program. A computer worm spreads (to a large extend) autonomously through a network thereby infecting computers with an autonomous copy of itself. Therefore, computer worms use known vulnerabilities in operating systems or software in order to infect a computer (see, e.g, [OEC09]).
Trojan Horse	A *Trojan horse* (or *Trojan*) is a malicious software program that tries to appeal to and interest the user by providing seemingly useful functionalities (or by actually offering a "trojanized" version of an existing useful program) (see, e.g., [OEC09]).
Exploit Kit	An *Exploit Kit* is a server application that is used to manage so-called *drive-by-download* attacks. A drive-by-download uses vulnerabilities in browsers and/or browser add-ons in order to automatically download malicious payload to a victim's computer as soon as the victim visits a compromised website. Thereby, no user interaction, such as clicking on buttons, is required (see, e.g, [KM13]).
Downloader	A *Downloader* is a type of malware that is typically very small and that per se does not perform any malicious actions. It is, however, used to download, extract and install malicious payload to the victim's computer when executed. Downloaders spread mostly through malicious attachments in phishing e-mails (see, e.g, [Szo05]).

Table 1: Malware Types

5 Interacting epidemics

In real life there is mostly not only one pathogen but rather a great number of pathogens spreading across a population at the same time (see, e.g., [AM91]). These pathogens might either co-exist without any point of contact or might interact in several possible ways (e.g. enhancing or weakening each other). These interactions are presented in further detail in the following sections.

5.1 Types of infections

In the scientific literature, a wide variety of terms is used to describe similar or equivalent cases of infections with two or more pathogens. Therefore, it is important to clearly define the different cases. Depending on the time of infection with the first and the second pathogen, one differentiates between a *subsequent infection*, *co-infection* and *superinfection*.[3]

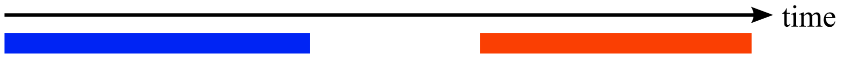


Figure 4: Subsequent Infection

In the case of *subsequent infections*, there are two possibilities, which are visualized in Figure 4 and Figure 5, respectively. In the first case (Figure 4), a host is primarily only infected with the first pathogen (blue). He then recovers completely from this infection and is subsequently infected with the second pathogen (orange) (see, e.g., [BS07]).

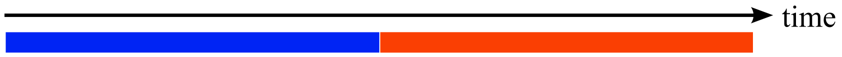


Figure 5: Hostile takeover

A special case of subsequent infections is visualized in Figure 5. In this case, a host is primarily only infected with the first pathogen (blue). Without recovering, however, a second pathogen (orange) "takes over" the host and suppresses the first pathogen. Due to the special behavior of the pathogens, throughout this thesis, this special case will be referred to as *hostile takeover*.

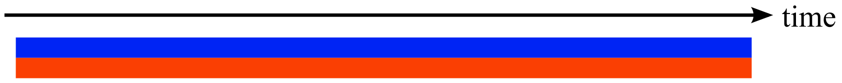


Figure 6: Co-Infection

[3] Note that these terms were primarily defined for HIV cases only, but were later also adapted to various other cases of infectious diseases (see, e.g., [SRL05]). Since they allow the most unambiguous distinction between the different cases they are also used within this thesis.

Possible cases of subsequent infections include *Cross-Immunity*, *Cross-Enhancement*, and *Hostile Takeover*. In the case of *cross-immunity*, a host primarily infected with one disease acquires short- or long-term immunity not only to this disease but also to a second one. This second disease might be either a different one or a competing strain of the same disease (see, e.g., [BvdDW08]). *Cross-Enhancement* (also known as *Antibody-Dependent Enhancement (ADE)*) is an effect that occurs in the case of multi-strain diseases. In particular, an individual recovered from an infection with one strain (serotype) acquires temporal cross-immunity against the other serotype(s). If, however, the level of protection decreases, subsequent infections with different serotypes are possible and the pre-existing antibodies actually increase the infectiousness of the individual (see, e.g., [WA14]). A *hostile takeover* can be observed between two competing (strains of) infectious diseases / malicious software programs. In this case, one infectious disease / malicious software program can take over a host that is already infected with a different disease / malware, i.e. the first disease / malware in the host is eliminated.

If a host simultaneously suffers from more than one disease, it is called a *dual infection*. Dual infections are subdivided into *co-infections* and *superinfections* (see, e.g., [SRL05], [BS07]). The term *co-infection* refers to cases in which the host is concurrently infected with two different (strains of) pathogens (see Figure 6). Thereby, it is required that the infection with both pathogens happens simultaneously or within a very short period of time (i.e. before an immune response to the first infection has developed). Co-infection might occur, for example, due to blood donations with two or more differently infected blood products (see, e.g., [SRL05], [BS07]).

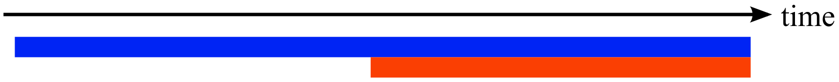


Figure 7: Superinfection

In the case of a *superinfection* (see Figure 7), a host is primarily only infected with the first pathogen (blue). During the course of the disease and with a substantial time lag (i.e. after an immune response to the first infection has developed), the host is then also infected with the second pathogen (orange).

Note that in the scientific literature the term *co-infection* is often used as an umbrella term for all cases of dual infections independent from the time of infection with the second pathogen. This might be justified by the fact that for certain cases it is not verifiable when the second infection occurred. In order to avoid confusion, however, in this thesis the terms *co-infection* and *superinfection* will be used according to the definitions given above and the term *dual infection* in order to refer to both.

Possible cases of dual infections include *Opportunistic infections*, *Hyperparasitic infections*, *Increased susceptibility & Increased Virulence*, and *Interference competition*. An *opportunistic infection* is an infection with a pathogen / malicious software program that has limited or no pathogenic effect (or cannot occur at all) in an otherwise healthy host, but can cause a serious and mostly progressing disease / malware infection in a host previously infected with a different primary disease / malicious software program (see, e.g., [Sym65]). *Hyperparasitism* describes an interaction between two parasites / malicious software programs and one host. In this case one parasite / malware parasitizes a second parasite / malware that in turn parasitizes a specific host type. This interaction is called *obligate hyperparasitism* if the first parasite / malware is not able to parasitize the host directly but only by parasitizing a second parasite / malware (see, e.g., [HH98]). In the case of *increased susceptibility*, an individual / a computer already infected with one disease / malware has an increased chance of getting infected with another disease / malware. The same effect, i.e. a higher chance of infection, can be also observed if the *virulence* of the pathogen is *increased*. This might be either caused by synergetic effects or by with-in host competition. In the case of with-in host competition two equally strong pathogens compete for the same hosts. This competition, however, triggers mutations, which increase the virulence of one of the two pathogens thereby allowing this pathogen to eventually obtain superiority over the other one (see, e.g., [GCLS$^+$07]). In case of *interference competition*, pathogens / malware programs adopt specific strategies "for directly inhibiting the growth, reproduction or transmission of competitors" [Mid09]. Interference competition can be observed for two different cases: In the first case, the host is only infected with the primary disease / malware, which prevents the host from dual infection with another disease / malware. In this sense, interference competition is comparable to cross-immunity, however, with the restriction that the immunity (or protection) lasts only as long as the infection with the primary disease / malware lasts. In the second case (also known as *counter-syndemics*), the host is first infected with the primary disease / malware and is then superinfected with a secondary disease / malware. This secondary disease / malware then suppresses the primary disease / malware as long as the secondary disease / malware lasts. After the host has recovered from the secondary disease / malware, the primary disease / malware will reactivate (see, e.g., [BdRSR06]).

6 Experiment

Increased susceptibility is very common in both the biological as well as the malware context and is responsible for the increased prevalence and spreading dynamics of certain infectious diseases and malicious software programs. Therefore, this interaction type is considered in more detail within this chapter. In particular, an existing model for increased susceptibility of infectious diseases is adjusted in order to meet the requirements to model two malicious software programs spreading across the same network.

The SIR × SIR Model was implemented in R using RStudio. Thereby, the R implementation of the simple epidemic model provided by the Institute of Integrative Biology of the ETH Zurich was used as the basic model [MÏ6]. Building upon this implementation, the SIR × SIR model was implemented in R using RStudio (for implementation details see Appendix A).

6.1 The SIR × SIR Model

Chen et al. proposed a simple SIR model (see Section 3.1.2) of fixed population size with the two diseases A and B [CGCG13]. They proposed that "the infection rate for disease A is increased, if the individual has or had disease B and vice versa" [CGCG13]. Hence, for each individual, there are nine possible states and two different infection rates. This is depicted as a flow chart with the nine states *S, A, B, AB, a, b, aB, Ab, ab* in Figure 8.

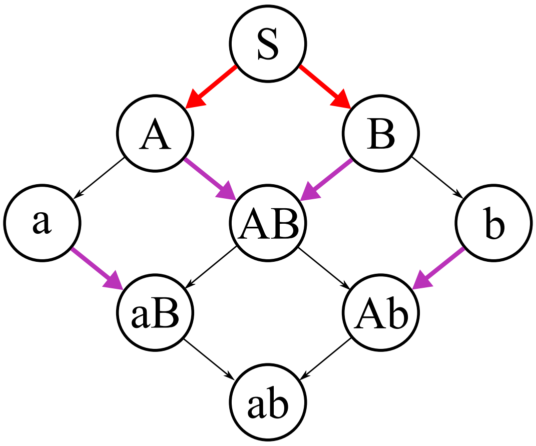


Figure 8: Flow Chart according to [CGCG13]

In order to fit this SIR × SIR model to the context of malware, several modifications were performed, which are visualized in Figure 9. First, a direct transition from the susceptible state *S* to the dually infected state *AB* was added in order to integrate co-infections into the model. This transition occurs with the normal infection rate α and is hence visualized as red arrow. Accordingly, also a direct recovery from the infected state *AB* to the recovered state *ab* was added. Second, the infection rate is only increased if the computer has malware A. This restriction was made for two reasons. First, if a computer system has recovered from an infection with immunity, e.g. by deleting the malware and protecting the system from future infections through an anti-malware software, this past infection does not have any influence on future infections. Hence, the transitions from states *a* and *b* to the states *aB* and *Ab* occur with the normal instead of the increased transmission rate (visualized by red arrows).

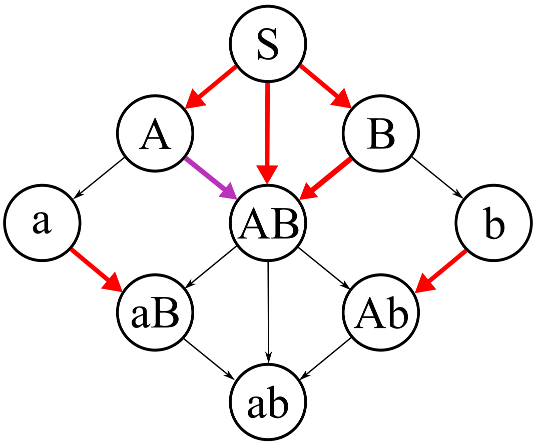


Figure 9: Modified Flow Chart based on [CGCG13]

Second, in most cases the interactions between two diseases or malicious software programs are not mutual, but rather unilateral. Accordingly, only the transition from state *A* to state *AB* occurs with the increased transition rate β (visualized by the purple arrow). The actual infection and recovery rates, which were used for the simulations, are depicted in Figure 10. The normal infection probability α was set to 0.13 (i.e. 13%) based on the average percentage of people who click on malicious links or open malicious attachments in phishing e-mails (see, e.g., [Ver16]). The increased infection probability β was set to 0.40 (i.e. 40%) based on the attack success rate of wide-spread exploit kits such as *Angler* (see, e.g. [Cis15]). The simulations were conducted with two different recovery rates, i.e. 0.04 and 0.14 in order to simulate infections that last for one week (1/7) as well as infections that last for four weeks (1/28).

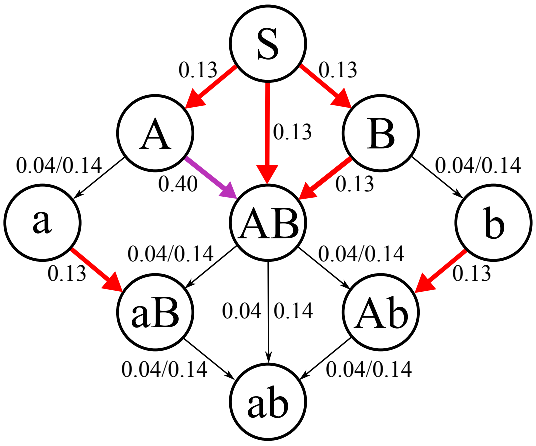


Figure 10: Flow Chart with Proposed Rates based on [CGCG13]

6.2 The example networks

For the simulations, eight different graphs were used, two of which represent real-life networks and six are generated graphs. In the following, the two real-life networks are presented in more detail and an overview of all graphs is given in Table 2 at the end of this section.

E-Mail Network

This graph represents the E-Mail communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain. The data for this graph was collected from January to March 2002. The 1,133 vertices represent users, i.e. university faculty, researchers, technicians, managers, administrators, and graduate students. Two vertices are connected by an edge if there is an E-Mail communication between them. This dataset was originally collected in order to investigate informal networks within organizations [KON16b]. The graph is an undirected, connected graph with 1,133 vertices and 5,451 edges.

Autonomous System

This graph represents a network of Autonomous Systems, which was constructed according to Border Gateway Protocol logs. The data was collected from the University of Oregon for the Route Views Project. The original dataset contains 733 daily instances which span an interval of 785 days from November 8 1997 to January 2 2000. For this thesis, only the graph with the largest number of vertices and edges (i.e. the dataset from January 02 2000) is used [KON16a].

Note that this graph originally comprised 13,895 edges including loops. Since loops, however, are irrelevant for the transmission of malware, they were eliminated from the graph using the R *simplify* function. The resulting graph consists of 6,474 vertices and 12,572 edges.

#	Name	Vertices	Edges	Min Degree	Max Degree	Diameter	Avg Path Length	Transitivity
1	E-Mails	1,133	5,451	1	71	8	3.61	0.166
2	$G(n, p)$ - 1	1,133	5,456	2	20	6	3.36	0.008
3	Scale-free - 1	1,133	5,650	5	110	5	3.06	0.028
4	Small-world - 1	1,133	5,665	7	12	10	5.39	0.570
5	Autonomous Systems	6,474	12,572	1	1460	9	3.71	0.010
6	$G(n, p)$ - 2	6,474	20,748	1	21	10	4.94	0.001
7	Scale-free - 2	6,474	12,945	2	118	9	5.23	0.002
8	Small-world - 2	6,474	12,948	1	7	33	16.74	0.425

Table 2: Example Graphs

6.3 The simulation parameters

While the infection probabilities were kept constant, the selection of the start vertices as well as the number of initially infected vertices and the recovery rate were modified for different simulation runs. In particular, the starting vertices were either selected randomly, or according to their eigenvector centrality, or according to their betweenness centrality (see Section 2.6).

#	Selection of Start Vertices	Number of Initially Infected Vertices	Recovery Rate
1	Random	1	0.04
2	Random	1%	0.04
3	Random	1	0.14
4	Random	1%	0.14
5	Eigenvector Centrality	1	0.04
6	Eigenvector Centrality	1%	0.04
7	Eigenvector Centrality	1	0.14
8	Eigenvector Centrality	1%	0.14
9	Betweenness Centrality	1	0.04
10	Betweenness Centrality	1%	0.04
11	Betweenness Centrality	1	0.14
12	Betweenness Centrality	1%	0.14

Table 3: Simulation constellations

The number of initially infected vertices was either one or one percent of the vertices in the respective graph. As mentioned in Section 6.1, the recovery rate varied between 0.04 and 0.14. These variations resulted in 12 distinct configurations, which are summarized in Table 3. Each of the 12 simulation configurations was run on each of the eight graphs resulting in 96 cases and 96 × 500 = 48,000 simulations.

6.4 Results

The results of the simulations are categorized into the prevalence, the incidence and the percentage of infected vertices. A detailed discussion and interpretation of the results is presented in section 6.5. All detailed results are summarized in Appendix B - G.

Prevalence

The prevalence describes the number of vertices in each compartment (*Susceptible, Infected, Recovered*) at each time step. Exemplarily, the prevalence of Simulation 1 of the E-Mail Graph (Graph 1) is depicted in Figure 11.

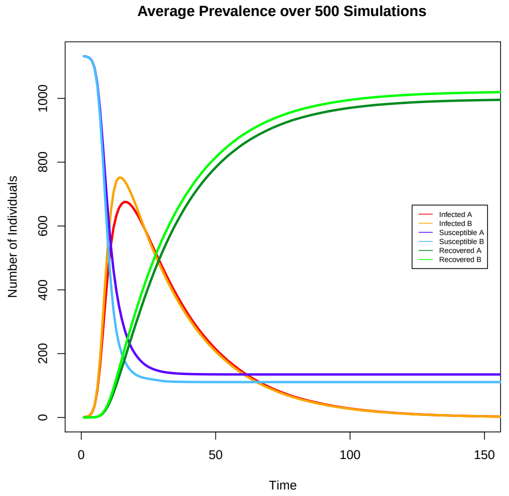


Figure 11: Prevalence: E-Mail Graph (Graph 1), Simulation 1

The prevalence is a good measure for getting an overview of the course and the duration of an epidemic outbreak. The prevalence is used to determine *the duration of the epidemic outbreak*, *the number of simultaneously infected vertices*, and *the time steps until the maximum number of simultaneously infected vertices is reached.*

Duration of the epidemic outbreak:

- For Graphs 1-7, the duration of the epidemic outbreak was solely determined by the recovery rate.
- For Graphs 1-7, the duration of the epidemic outbreak differed only marginally between Malware A and B, with Malware B lasting, in most cases, only between 0-5 time steps shorter than Malware A.
- For Graphs 1-3, the duration of the epidemic outbreak lasted between 175-182 time steps if the recovery rate was equal to 0.04 and between 51-60 time steps if the recovery rate was equal to 0.14. The difference between Malware A and B ranged between 0-3 time steps.

- For Graph 4, the duration of the epidemic outbreak lasted between 181-192 time steps (recovery rate 0.04) and 59-78 time steps (recovery rate 0.14), respectively. The difference between Malware A and B ranged between 1-7 time steps.
- For Graphs 5-7, the duration of the epidemic outbreak lasted between 209-250 time steps (recovery rate 0.04) and between 62-92 time steps (recovery rate 0.14).
- For Graph 8, the duration of the epidemic outbreak lasted between 263-385 time steps (recovery rate 0.04). The difference between Malware A and B was significantly larger than for Graphs 5-7.
- For Graph 8, the epidemic would not take off (i.e. it would reach only a very small percentage of vertices before dying out) if the recovery rate was set to 0.14.

Number of simultaneously infected vertices:

- Overall, the $G(n, p)$ random graphs (Graphs 2 & 6) and the scale-free graphs (Graphs 3 & 7) showed the highest peak numbers of simultaneously infected vertices. Moreover, they also proved to have the highest differences between Malware A and B.
- For Graphs 1,2,3 and 6 the outcome was largely determined by the recovery rate if the initially infected vertex/vertices was/were selected based on either the Eigenvector centrality or the Betweenness centrality. In contrast, if the initially infected vertex/vertices was/were selected randomly the outcome was higher if the number of initially infected vertices was equal to 1% as compared to those cases, in which the number of initially infected vertices was equal to 1.
- Graphs 5 and 7 showed very similar results, with the distinction, however, that the peak number of simultaneously infected vertices was considerably smaller for Simulation 3 than for all other cases. Moreover, the difference between Malware A and B was significantly smaller for Simulation 3 than for all other simulations.
- Graphs 4 and 8 (the small-world graphs), however, showed an overall lower maximum number of simultaneously infected vertices.

Time steps until maximum number of simultaneously infected vertices is reached:

- Graphs 1-3 and Graphs 6 & 7, respectively showed very similar results. In comparison, it took significantly longer on the two small-world graphs (Graphs 4 & 8) to reach the maximum number of simultaneously infected vertices.
- For epidemics spreading across the Autonomous Systems graph (Graph 5) the maximum number of simultaneously infected vertices was reached extremely fast and showed results comparable to the small scale-free graph (Graph 3).
- For all graphs except of the small-world graphs (Graphs 4 & 8) it took considerably longer to reach the maximum number of simultaneously infected vertices for Simulations 1 and 3 (i.e. if the initially infected vertex is chosen randomly).

- Whereas the difference between Malware A and Malware B was consistently between 0-3 time steps for Graphs 1-3 and Graphs 5-7, the difference was between -2 and 16 time steps for the small-world graphs (Graphs 4&8). The negative differences for the epidemics spreading across Graph 8 were all associated with a recovery rate of 0.14.

Incidence

The incidence describes the number of newly infected vertices per time step. Exemplarily, the incidence of Simulation 1 of the E-Mail Graph (Graph 1) is depicted in Figure 12.

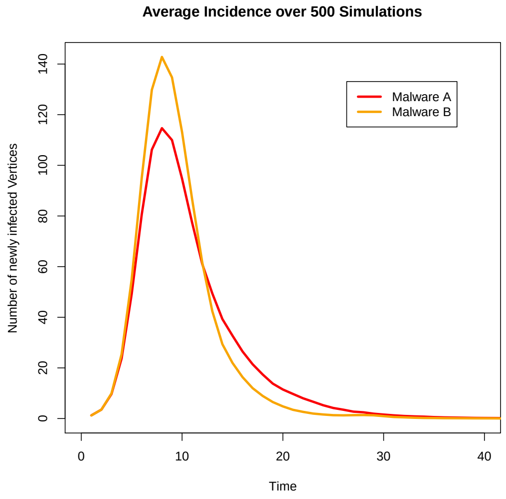


Figure 12: Incidence: E-Mail Graph (Graph 1), Simulation 1

For the analysis of the results, *the maximum incidence* as well as *the number of time time steps until no new infections occur* (i.e. the duration of the spreading) were considered:

Maximum Incidence:

- All four smaller graphs (Graphs 1-4) showed significantly higher percentages of maximum incidences than the larger graphs (Graphs 5-8).
- For all eight graphs, the difference between Malware A and Malware B reached a significant low point for Simulation 3 (i.e. one starting vertex is randomly selected and the recovery rate is equal to 0.14). Moreover, the difference between Malware A and B proved to be consistently smaller for the two small-world graphs (Graphs 4& 8) than for all the other graphs.
- For epidemics spreading across all graphs except of the small-world graphs the following pattern applied: If the initially infected vertex/vertices is/are selected based on either the Eigenvector centrality or the Betweenness centrality, then the number of initially infected vertices has only minor influence on the outcome and the recovery rate has major influence on the outcome. If, however, the initially infected vertex/vertices are selected

randomly, then the number of initially infected vertices as well as the recovery rate have major influence on the outcome.

Time steps until maximum incidence is reached:

- The epidemics spreading across the generated graphs (Graphs 2-4 and Graphs 6-8) showed a consistent pattern, i.e. the maximum incidence was reached fastest for the scale-free graphs (Graphs 3 & 7) followed by the $G(n, p)$ random graphs (Graphs 2 & 6) and slowest for the small-world graphs (Graphs 4 & 8)[4]. Thereby, it took significantly more time steps to reach the maximum incidence on the small-world graphs than on the scale-free graphs (i.e. up to ten times as long).
- In case of the E-Mail Graph (Graph 1), the number of required time steps was almost equal to the one of the small scale-free graph (Graph 3). In case of the Autonomous Systems Graph (Graph 5), however, the maximum incidence was reached much faster than in any other case.
- Regarding the difference between Malware A and Malware B, we found that for the epidemics spreading across the Graphs 1 - 4, 6, and 7 the difference remained very small ranging only between 0 and 2 time steps. Differing behavior was only observed for the Graphs 5 & 8. In case of Simulations 6,8,10, and 12 the difference between Malware A and Malware B was always equal to -1 (meaning that the maximum incidence was reached slower for Malware B) for epidemic spreading across the Autonomous Systems Graph (Graph 5). For epidemics spreading across the small-world Graph (Graph 8), the difference between Malware A and B was significantly larger than for all the other graphs ranging between 3 and 17 time steps[5].

Percentage of Infected Vertices

The overall percentage of infected vertices during the epidemic outbreak was calculated as the average percentage of vertices in the population that has been infected with Malware A (or B, respectively) within the time frame from time step t_0 until no new infections occurred. When analyzing these results, we found that:

- Overall, the percentages of infected vertices was significantly higher for the smaller graphs (Graphs 1 - 4) than for the larger graphs (Graphs 5 - 8).
- The significantly highest percentage of infected vertices for all larger graphs was observed for the $G(n, p)$ Random Graph (Graph 6).
- Overall, the generated gaphs (Graphs 2 - 4 and Graphs 6 - 8) showed significantly higher percentages of infected vertices than the real-world graphs (Graphs 1 & 5).

[4] With the exception of those cases for the small-world Graph 8, in which the epidemic did not spread (Simulations 3,4,7,8,11,12).

[5] Excluding those cases in which the epidemic did not spread (Simulations 3,4,7,8,11,12).

- For all eight graphs, there was no significant difference in the outcome between those simulations, in which the starting vertex/vertices was/were selected based on the Eigenvector centrality (Simulations 5 - 8) and those, in which the starting vertex/vertices was/were selected based on the Betweenness centrality (Simulations 9 - 12).
- Overall, the number of initially infected vertices did not have a significant influence on the outcome when the selection criterion for the starting vertex/vertices was either the Eigenvector centrality (Simulations 5 - 8) or the Betweenness centrality (Simulations 9 - 12).
- In all cases except of the small-word graphs (Graphs 4 and 8) there was a significant difference in the outcome between those simulations, in which the starting vertex/vertices was/were selected randomly (Simulations 1 - 4) and those simulations, in which the starting vertex/vertices was/were selected based on either of the two centrality measures (Simulations 5 - 12).
- All eight graphs showed a significant difference in the outcome between those simulations, in which the recovery rate was set to 0.04 and those, in which the recovery rate was set to 0.14. This difference was significantly higher for most larger graphs (Graphs 5 - 8) than for the smaller graphs (Graphs 1 - 4), with the $G(n, p)$ Random Graph (Graph 6) being the exception.

6.5 Discussion

The difference in the duration of the epidemic outbreak between the synthetic small-world graphs (Graphs 4 & 8) and the other graphs (Graphs 1-3 & 5-7) can be best explained by their unique combination of transitivity, the average path length, and the diameter of the respective graphs. Although the long-range connections in small-world graphs facilitate and accelerate the spreading dynamics of epidemics in comparison to regular graphs (see, e.g., [WS98], [KE05]), the high clustering and the relatively low number of long-range connections lead to a distinctively slower spreading behavior in comparison to the other graphs. Hence, both Malware A and Malware B need a considerably longer time to spread through the small-world graphs (Graphs 4 & 8) than through the other graphs (Graphs 1-3 & 5-7). A higher recovery rate additionally impedes the spreading dynamics of the epidemics, which results, hence, in even lower numbers of simultaneously infected vertices for the small-world graphs (Graphs 4 & 8) (see, e.g, [KE05], [Lew09]). Also, the larger difference between Malware A and Malware B in comparison to the other graphs might be explained by this specific spreading behavior. In particular, the increased transmission probability β allows Malware B to spread faster via the long-range connections once the connected vertex is infected with Malware A (in comparison, Malware A does not have an increased transmission probability).

Due to this high clustering and the sparse long-range connections, epidemics are also more likely to die out quickly (i.e. to not spread through the graph) in a small-world graph than in the other observed graph topologies. Especially in the case of a higher recovery probability, the chance that the epidemics spread only within the cluster(s) in which the initially infected vertices are positioned in, is significantly increased, because the vertices, which might infect another vertex via a long-range connection are more likely to recover before they can transmit the malware to another cluster.

Furthermore, Lewis showed that due to these distinct graph properties, small-world graphs exhibit a distinctively smaller *spectral radius*[6] than other graph topologies [Lew09]. This is insofar relevant as Wang et al. proved that the epidemic threshold is equal to

$$\tau = \frac{1}{\lambda_{dom}} \tag{6.1}$$

where λ_{dom} is the dominant Eigenvalue of the adjacency matrix (for further details and the full mathematical proof see, e.g., [WCWF03], [VMOJK09]). Based on these findings, it follows that the smaller the spectral radius of a graph, the lower the probability of an epidemic to spread through this graph (see, e.g., [WCWF03], [JKVMVD06], [Lew09], [CWW$^+$08]). As can be seen in Table 4, the spectral radius of Graph 8 is significantly smaller than the spectral radii of Graphs 1-7, which helps to explain why the epidemics do not spread across Graph 8 for a recovery probability of 0.14.

Graph	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**
Spectral Radius	20.747	10.688	20.376	10.088	46.318	7.579	12.434	4.149

Table 4: Spectral radii

Furthermore, Table 4 reveals that the Autonomous Systems graph (Graph 5) has a considerably larger spectral radius than the other graphs. Since a lower spectral radius lowers the spreading probability of an epidemic, a larger spectral radius increases the spreading probability. This might explain why the maximum number of simultaneously infected vertices and the maximum incidence was reached significantly faster on Graph 5 than on the other larger graphs.

The larger difference between Malware A and Malware B can be also interpreted as a result of this accelerated spreading behavior, because a larger number of vertices infected with Malware A offers more possibilities (with increased transmission probability β) for Malware B to spread through the graph. This means that in the case of $G(n, p)$ random graphs and scale-free graphs the increased infection probability due to previous infection with Malware A acts as an amplifier for the spreading of Malware B.

Furthermore, the results showed that the vertex with the highest centrality measure (equally true for the Betweenness centrality and the Eigenvector centrality) has the highest influence on the outcome and that adding more vertices based on any of the selected centrality measures has only relatively small effects on the peak number of simultaneously infected vertices and the

[6] The *spectral radius* of a graph G is defined as the dominant Eigenvalue of the graph's adjacency matrix [Lew09].

maximum incidence. Consequently, the higher number of initially infected vertices has also only minor effects on the percentage of overall infected vertices.

In contrast, the results for Simulations 1 & 3 (i.e. those simulations, for which only one initially infected vertex was selected randomly) showed significantly lower results for the peak number of simultaneously infected vertices, the maximum incidence, and the percentage of overall infected vertices, and higher results for the required number of time steps until the peak number of simultaneously infected vertices is reached. This might be, because in the case of 1% of initially infected vertices the chances are much higher that a vertex with a high centrality score is among the initially infected ones than if only a single vertex is randomly selected as the initially infected one. As described in Section 6.4, however, Simulations 1 & 3 result mostly in a higher difference between Malware A and Malware B with respect to the peak number of simultaneously infected vertices than the other simulations. This can be explained by the random selection criteria of the initially infected vertex, which mostly results in two different initially infected vertices (i.e. one vertex infected with Malware A and a different vertex infected with Malware B). This means that the two epidemics start to spread independently and result in super-infections (see Section 5.1) once one of the epidemics is transmitted to a vertex that is already infected with the other epidemic. Once Malware B reaches the cluster of vertices already infected with Malware A, the increased transmission probability β significantly accelerates the spreading of Malware B among those vertices that are currently infected with Malware A. In comparison, Malware A needs considerably longer to super-infect those vertices, which are currently infected with Malware B, which means that more vertices also recover from Malware A during this time. Hence, this results in a comparatively high difference between Malware A and Malware B. At the same time, this means that both Malware A and Malware B show a relatively similar maximum incidence, which results hence in a very small difference between Malware A and Malware B. In case of Simulations 5, 7, 9, and 11, on the other hand, both epidemics always start from a single co-infected vertex (see Section 5.1). This means that both Malware A and Malware B start to spread similarly from the initially co-infected vertex and that there is a smaller number of vertices only infected with Malware A that can act as accelerator for Malware B.

In case of a recovery rate of 0.14, however, the number of initially infected vertices had a significantly higher influence on the number of simultaneously infected vertices than in the case of a recovery rate of 0.04. This was equally true for all simulations independent of the selection criteria for the initially infected vertex/vertices. This is, because a small recovery rate such as 0.04 allows an epidemic to spread almost unimpeded through a graph. The very high percentages (i.e. up to 100%) of overall infected vertices for all graphs in the simulations with a recovery rate of 0.04 (Simulations 1, 2, 5, 6, 9, 10) demonstrate this unimpaired spreading behavior. Hence, the larger number of initially infected vertices has only a small accelerating effect on an epidemic. In case of a significantly higher recovery rate such as 0.14, however, an epidemic is considerably slowed down in its spreading dynamic and results hence in a much shorter duration of the epidemic outbreak and a smaller percentage of overall infected vertices. Accordingly, an epidemic that starts from only one initially infected vertex reaches a considerably smaller percentage of vertices and results in a smaller number of simultaneously infected vertices. Starting, however, with 1$\%$ of initially infected vertices (in comparison to only one initially infected vertex) counteracts this decelerated spreading behavior and results hence in a higher number of simultaneously infected vertices after fewer time steps and a higher percentage of overall infected vertices[7].

Graph	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**
Edge Density	0.0085	0.0085	0.0088	0.0088	0.0006	0.0009	0.0006	0.0006

Table 5: Edge Density

As described in Section 6.4 the epidemics reached higher percentages of infected vertices in the smaller graphs (Graphs 1-4) then in the larger graphs (Graphs 5-8). This is, because given the same transmission and recovery probability, an epidemic will reach a higher percentage of vertices in a smaller and better connected graph (i.e. a graph with a higher *edge density*[8]) than in a larger and sparser connected one. Since Graphs 1-4 are about six times smaller (considering the number of vertices) than Graphs 5-8 and have a much higher edge density (see Table 5), both Malware A and Malware B reached a considerably higher percentage of the vertices in the smaller graphs (Graphs 1-4) than in the larger graphs (Graphs 5-8). The higher edge density (see Table 5) of the smaller graphs (Graphs 1-4) also led to a higher maximum incidence than for the larger graphs (Graphs 5-8).

Moreover, the larger $G(n, p)$ random graph (Graph 6) has a notably higher edge density than the other larger graphs (Graph 5,7,8, see Table 5). This property in connection with the fact that epidemics, in general, spread very fast through $G(n, p)$ random graphs (see, e.g., [WS98], [KE05]) can be seen as the reason why both Malware A and Malware B reached a considerably higher percentage of the vertices in Graph 6 compared to the other larger graphs (Graphs 5,7,8) and also explains why the higher recovery rate of 0.14 had a smaller influence on Graph 6 than on the other larger graphs (Graphs 5,7,8). This is also in line with the result that the Graphs 5 and 7 showed a significantly smaller peak number for Simulation 3 than Graph 6 (see Section 6.4)[9].

[7] This is true for all graphs except of the larger small-world graph (Graph 8), because, as described above, the epidemics would not take off for those simulations, where the recovery rate is equal to 0.14. For Malware B to have a spreading advantage due to an increased transmission probability, however, a substantial number of vertices must be infected with Malware A. Since this requirement is not fulfilled for these simulations, the difference between Malware A and Malware B (in terms of required time steps to reach the peak number of simultaneously infected vertices) is close to zero or even negative, which means that Malware A reached the peak number faster than Malware B. Similarly, the peak numbers of simultaneously infected vertices are almost equal as is also the duration of the epidemic outbreak.

[8] The *edge density* is defined as the ratio of the number of edges and the number of possible edges excluding loops and was calculated using igraph's *edge_density* function.

[9] Since both Malware A and Malware B did not take off in Simulation 3 on the small-world graph (Graph 8), the results are not comparable to the Graphs 5, 6, and 7.

7 Conclusion

The literature research has shown that in the case of interacting epidemics one can differentiate between subsequent infections and dual infections. Whereas subsequent infections are very common in the biological context and lead, among others, to either cross-immunity or cross-enhancement, these cases are (at least currently still) irrelevant for the context of computer malware. In contrast, hostile takeovers can be repeatedly observed in the context of computer malware and are rather rare in the biological context. In case of dual infections, all interaction types can be observed in both the biological as well as the malware context. Thereby, especially increased susceptibility (and increased virulence) is a very common case in both fields.

Based on these observations, an existing model for increased susceptibility of infectious diseases was selected and modifications were applied in order to fit this model to the context of computer malware. The selected model was an SIR $\times$ SIR compartmental model, which means that each vertex in the network can be in one of nine stages. The major modifications included the addition of a direct infection with both malicious software programs (enabling hence co-infections) and the restriction to a unilateral increase in the infection rate.

The adapted model was implemented in R using RStudio. Two real-life networks (a smaller and a larger one) were selected for the purposes of the experiment and six graphs synthetically generated on the basis of the two selected ones. Then, simulations were conducted using different parameter settings. In particular, the selection of the start vertices, the number of initially infected vertices, as well as the recovery rate were modified. These variations resulted in twelve distinct configurations, which were run on each of the eight graphs resulting hence in 96 cases and 96 $\times$ 500 = 48,000 simulations.

The results showed that whereas epidemics spreading across the real-life networks as well as the synthetically generated $G(n,p)$ random graphs and scale-free graphs showed surprisingly similar behavior, the epidemic spreading differed greatly when the epidemics spread through a synthetically generated small-world graph. These differences could be best explained by the unique combination of transitivity, average path length, and diameter of the respective graphs. Furthermore, the spectral radii of the graphs proved to be a valid measure for the spreading dynamics of the interacting epidemics, i.e. the smaller the spectral radius of a graph, the lower the probability of an epidemic to spread through this graph. In addition, it was confirmed that the final outcome, i.e. the final number of infected vertices, was dependent on the edge density of the graph. Finally, the results showed that the selection criterion of the initially infected vertex/vertices, has significant influence on the spreading behavior of the epidemics. In particular, the epidemics are more severe (i.e. they reach more vertices in the network) when the initially infected vertex/vertices is/are selected based on centrality measures than when it/they is/are selected randomly.

This thesis focused solely on the case of increased susceptibility, but did not apply existing models of other cases to the context of computer malware. Hence, future research could fill this research gap and run similar simulations using, however, other models.

Bibliography

[AM91] Roy M. Anderson and Robert M. May. *Infectious Diseases of Humans*. Oxford University Press, First edition, 1991.

[Bai75] Norman T.J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Charles Griffin & Company Ltd, Second edition, 1975.

[Bar14] Albert-László Barabási. *Network Science*. Cambridge University Press, 2014.

[BdRSR06] A.S. Bell, J.C. de Roode, D. Sim, and A.F. Read. Within-host competition in genetically diverse malaria infections: parasite virulence and competitive success. *Evolution*, 60(7):1358–1371, 2006.

[Bon72] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2(1):113–120, 1972.

[BS07] Jason T. Blackard and Kenneth E. Sherman. Hepatitis C Virus Coinfection and Superinfection. *The Journal of Infectious Diseases*, 195(4):519–524, 2007.

[BvdDW08] Fred Brauer, Pauline van den Driessche, and Jianhong Wu, editors. *Mathematical Epidemiology*. Springer-Verlag Berlin, First edition, 2008.

[CFG$^+$01] P. Crepel, S.E. Fienberg, J. Gani, C.C. Heyde, and E. Seneta, editors. *Statisticians of the Centuries*. Springer-Verlag New York, First edition, 2001.

[CGCG13] Li Chen, Fakhteh Ghanbarnejad, Weiran Cai, and Peter Grassberger. Outbreaks of coinfections: the critical role of cooperativity. *EPL (Europhysics Letters)*, 104(5):1234–1238, 2013.

[Cis15] Cisco Systems Inc. Cisco 2015 Midyear Security Report. Technical report, 2015.

[Coo10] P. Coogan. SpyEye Bot versus Zeus Bot. Online source: http://www.symantec.com/connect/blogs/spyeye-bot-versus-zeus-bot, 2010. (Accessed: 10.07.2017).

[Cor15] IBM Corporation. Meet Shifu. Masterful New Banking Trojan Targeting Japan and UK banks. Version 1.0 September 2015. Technical report, IBM Corporation, 2015.

[Cor16] Symantec Corporation. Internet Security Threat Report. United States of America, April 2016.

[CWW$^+$08] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security*, 10(4):1–26, 2008.

[GCLS$^+$07] Andrea L. Graham, Isabella M. Cattadori, James O. Lloyd-Smith, Matthew J. Ferrari, and Ottar N. Bjørnstad. Parasite adaptations to within-host competition. *Trends in Parasitology*, 23(6):284–291, 2007.

[Gem97] Wilhelm Gemoll. *Griechisch-Deutsches Schul- und Handwörterbuch*. Publisher Hölder-Pichler-Tempsky, 9th edition, 1997.

[Her07] Simon Heron. Cyber Gangs: Gang culture in the online world. *Network Security*, 2007(11):4–7, 2007.

[Het89] Herbert W. Hethcote. Three basic epidemiological models. In Simon A. Levin, Thomas G. Hallam, and Louis J. Gross, editors, *Applied Mathematical Ecology*, chapter 3, pages 119–144. Springer Berlin Heidelberg, 1989.

[HH98] Robert D. Holt and Michael E. Hochberg. The Coexistence of Competing Parasites. Part II - Hyperparasitism and Food Chain Dynamics. *Journal of Theoretical Biology*, 193(3):485–495, 1998.

[JKVMVD06] A. Jamakovic, R.E. Kooij, P. Van Mieghem, and E.R. Van Dam. Robustness of networks against viruses: The role of the spectral radius. *Proceedings of the 13th Annual Symposium of the IEEE/CVT Benelux*, Proceedings:35–38, 2006.

[KCW93] Jeffrey Kephart, David Chess, and Steve White. Computers and epidemiology. *IEEE Spectrum*, 30(5):1–10, 1993.

[KE05] Matt J. Keeling and Ken T.D. Eames. Networks and epidemic models. *Journal of the Royal Society Interface*, 22(4):295–307, 2005.

[KM13] Vadim Kotov and Fabio Massacci. Anatomy of Exploit Kits. Preliminary Analysis of Exploit Kits as Software Artefacts. In Jan Jürjens, Benjamin Livshits, and Riccardo Scandariato, editors, *Engineering Secure Software and Systems*, chapter 13, pages 181–196. Springer Berlin Heidelberg, 2013.

[KON16a] KONECT. Route views. Online source: http://konect.uni-koblenz.de/networks/as20000102, 2016. (Accessed: 07.11.2016).

[KON16b] KONECT. U. rovira i virgili network dataset. Online source: http://konect.uni-koblenz.de/networks/arenas-email, 2016. (Accessed: 07.11.2016).

[Lew09] Ted G. Lewis. *Network Science: Theory and Practice*. John Wiley & Sons, Inc, 2009.

[M1̈6] Viktor Müller. Network models of epidemics. Online source: http://www.tb.ethz.ch/education/learningmaterials/modelingcourse/level-2-modules/network.html, 2016. (Accessed: 01.11.2016).

[McL00] Kari S. McLeod. Our sense of Snow: the myth of John Snow in medical geography. *Social Science & Medicine*, 50:923–935, 2000.

[McM10] R. McMillan. New Russian botnet tries to kill rival. Online source: http://www.computerworld.com/article/2520797/malware-vulnerabilities/new-russianbotnet- tries-to-kill-rival.html, 2010. (Accessed: 10.07.2017).

[Mid09] Nicole Mideo. Parasite adaptations to within-host competition. *Trends in Parasitology*, 25(6):261–268, 2009.

[Mil67] Stanley Milgram. The Small-World Problem. *Psychology Today*, 1(1):61–67, 1967.

[MK14] J. C. Miller and I. Z. Kiss. Epidemic Spread in Networks: Existing Methods and Current Challenges. *Mathematical Modelling of Natural Phenomena*, 9(2):4–42, 2014.

[MMG06] Paul Martin and Estelle Martin-Granel. 2,500-year Evolution of the Term Epidemic. *Historical Review*, 12(6):976–980, 2006.

[New12] M. E. J. Newman. *Networks. An Introduction.* Oxford University Press, 2012.

[OEC09] OECD. *Computer Viruses and Other Malicious Software. A Threat to the Internet Economy.* OECD publishing, 2009.

[Org15] World Health Organization. Health in 2015: from MDGs, Millennium Development Goals to SDGs, Sustainable Development Goals. France, 2015.

[SRL05] Davey M. Smith, Douglas D. Richman, and Susan J. Little. HIV Superinfection. *The Journal of Infectious Diseases*, 192(3):438–444, 2005.

[Sym65] W.S. Symmers. Opportunistic infections. The concept of 'opportunistic infections' . *Proceedings of the Royal Society of Medicine*, 58:341–346, 1965.

[Szo05] Peter Szor, editor. *The Art of Computer Virus Research and Defense.* Addison Wesley Professional, First edition, 2005.

[Ver16] Verizon. 2016 Data Breach Investigations Report. Technical report, 2016.

[VMOJK09] Piet Van Mieghem, Omic, Jasmina, and Robert Kooij. Virus spread in networks. *IEEE/ACM Transactions on Networking*, 17(1):1–14, 2009.

[vS10] Maarten van Steen. *Graph Theory and Complex Networks. An Introduction.* Maarten van Steen, 2010.

[WA14] Hannah Woodall and Ben Adams. Partial cross-enhancement in models for dengue epidemiology. *Journal of Theoretical Biology*, 351:67–73, 2014.

[WCWF03] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: an eigenvalue viewpoint. *22nd International Symposium on Reliable Distributed Systems*, Proceedings:25–34, 2003.

[WS98] Duncan Watts and Steven Strogatz. Collective dynamics of small-world networks. *Nature*, 6684(393):440–442, 1998.

Appendix A

Initialization

```
# Set the number of initially infected individuals
init_inf_A = 1
init_inf_B = 1

# Set the normal infection rates
alpha_A = 0.13
alpha_B = 0.13

# Set the increase of the infection rate for disease A and B
beta_A = 0
beta_B = 0.27

# Set the recovery rates
gamma_A = 0.14
gamma_B = 0.14

# Set the length of the simulation
simlength = 70
# Set how often the simulation is repeated
simnumber = 500

# Set TRUE for plotting the graph
plot.spread = FALSE

# Set whether the infected individuals can recover
# from timestep 0 to timestep 1
recovery.wait_A = TRUE
recovery.wait_B = TRUE
```

Listing 1: Initialization

Selection of Edges

```
for (k in 1:simlength) {

    # 4 lists in order to sort the edges according
    # to their transmission risk
    highrisk.edges_A = list()
    highrisk.edges_B = list()
    normal.edges_A = list()
    normal.edges_B = list()

    [...]

     for(i in 1:nrow(edgelist)){

      # Find those vertices, which are
      # ONLY INFECTED WITH DISEASE A
      if(infected_A[edgelist[i,1]] %in% TRUE &&       infected_B[
         edgelist[i,1]] %in% FALSE){

        # Find those vertices, which are
        # ONLY INFECTED WITH DISEASE B
        if(infected_A[edgelist[i,2]] %in% FALSE &&      infected_B[
           edgelist[i,2]] %in% TRUE){

          # The variable n and o are "counters"
          # and are used to append the value to the lists
          highrisk.edges_A[[n]] = i
          highrisk.edges_B[[o]] = i
          n = n + 1
          o = o + 1
          next
        }

        [...]
```

Listing 2: Edge sorting

Transmission and Recovery

```r
# The rbinom function is used to determine those edges along which the disease is transmitted
# Differentiation between the normal transmission probability and the increased transmission probability
transmit_normal_A = rbinom(length(normal.edges_A),1,alpha_A)
transmit_high_A = rbinom(length(highrisk.edges_A),1,          (alpha_A+beta_A))

# The edges along which the diseases are transmitted are selected
transmitter.edges_normal_A =
  normal.edges_A[transmit_normal_A == 1]
transmitter.edges_high_A =
  highrisk.edges_A[transmit_high_A == 1]

# Based on the selected edges the corresponding vertices are selected
vertices.transmitter.edges_normal_A = unique(as.vector(edgelist[transmitter.edges_normal_A,1:2]))
vertices.transmitter.edges_high_A = unique(as.vector(edgelist[transmitter.edges_high_A,1:2]))

# All vertices, which are now newly infected are set to TRUE in the corresponding logical vector
infected_A[vertices.transmitter.edges_normal_A] = TRUE
infected_A[vertices.transmitter.edges_high_A] = TRUE
```

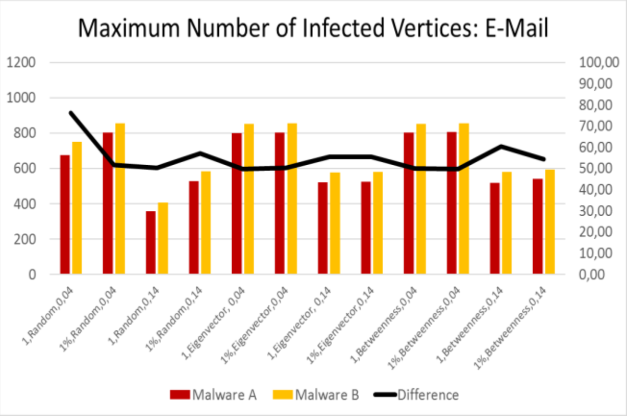
Listing 3: Transmission

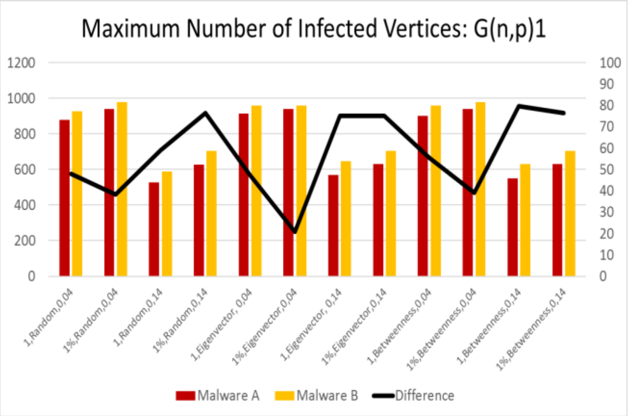
```r
# Depending on recovery.wait
# First, those vertices which are infected with disease A are selected
# Then, those vertices, which recover during this time step are selected with probability gamma
# Finally, the recovered vertices are set to NA
if (recovery.wait_A == FALSE || k > 1){
  infected.vertices_A = which(infected_A %in% TRUE)
  recover_A =
    rbinom(sum(infected_A %in% TRUE, na.rm = TRUE),1,gamma_A)
  recover.vertices_A = infected.vertices_A[recover_A == 1]
  infected_A[recover.vertices_A] = NA
}
```

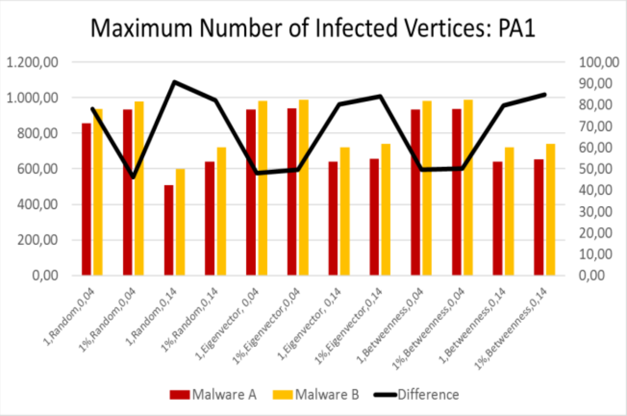
Listing 4: Recovery

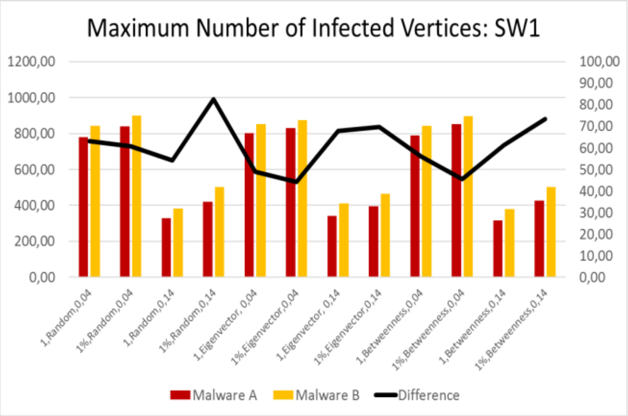
APPENDIX B

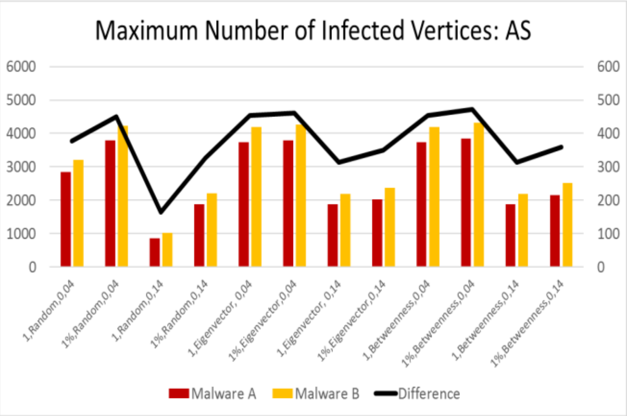
Prevalence – Maximum Number of Infected Vertices

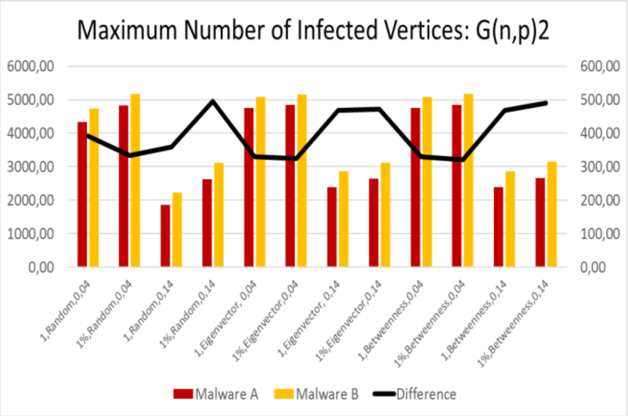

Maximum Number of Infected Vertices: E-Mail

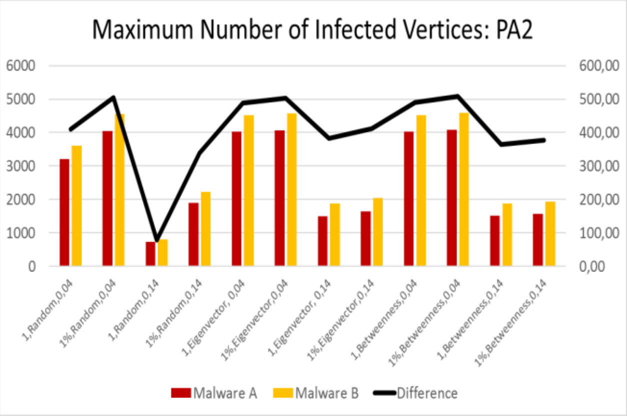

Maximum Number of Infected Vertices: G(n,p)1

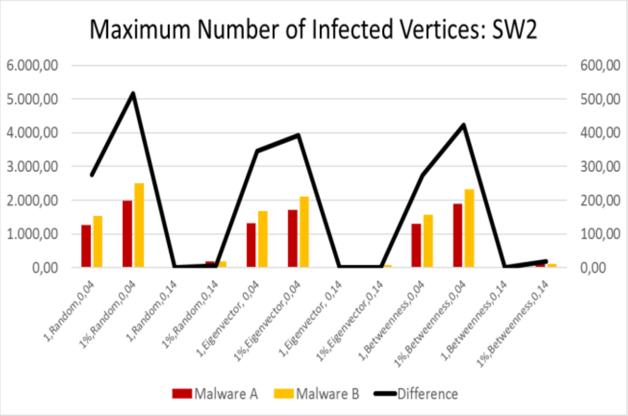

Maximum Number of Infected Vertices: PA1


Maximum Number of Infected Vertices: SW1


Maximum Number of Infected Vertices: AS

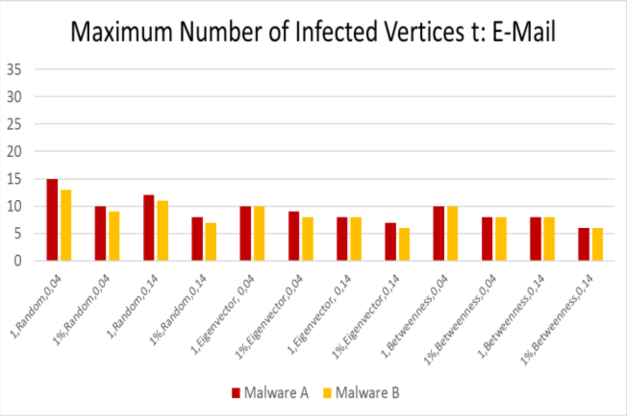

Maximum Number of Infected Vertices: G(n,p)2

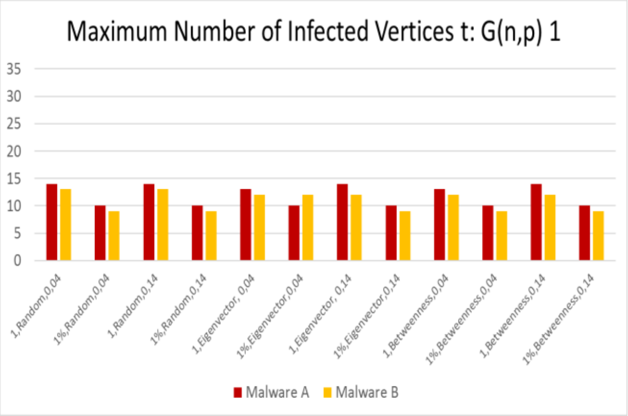

Maximum Number of Infected Vertices: PA2

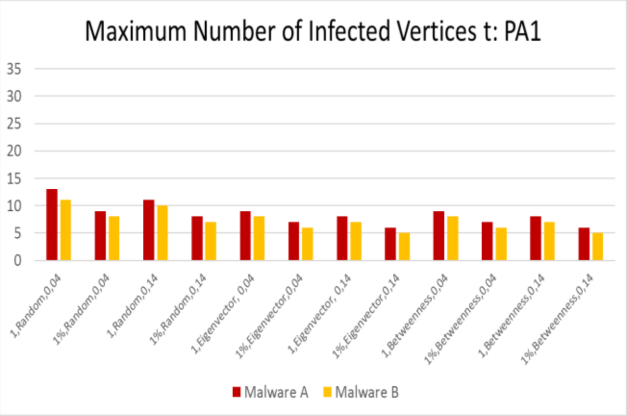

Maximum Number of Infected Vertices: SW2

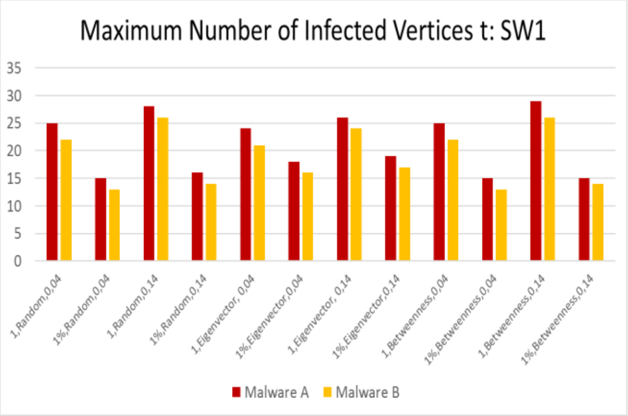
APPENDIX C

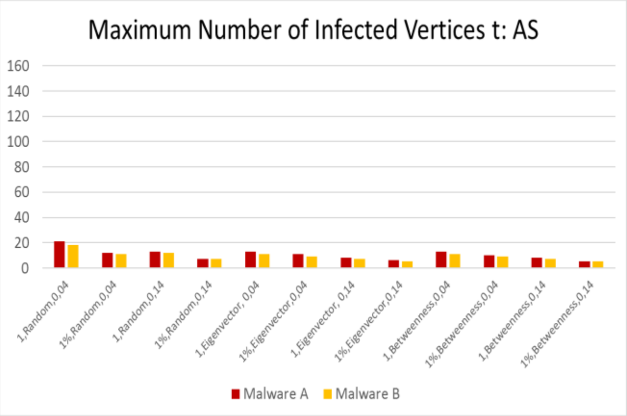
Prevalence – Time until Maximum Number of Infected Vertices is reached

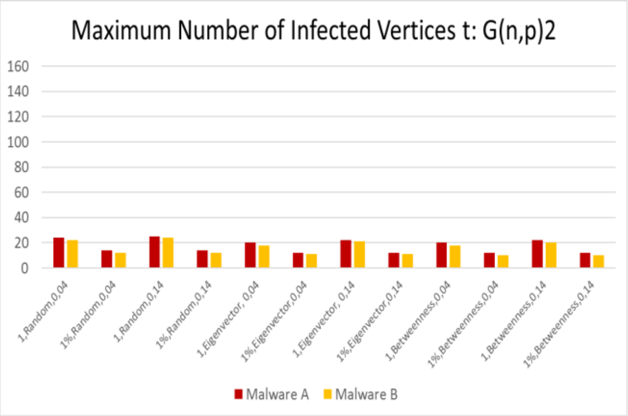


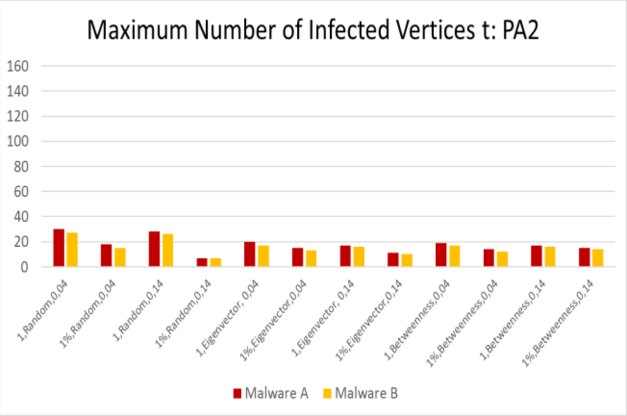


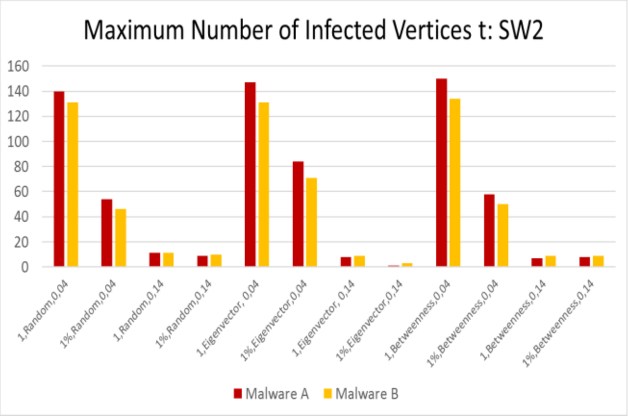






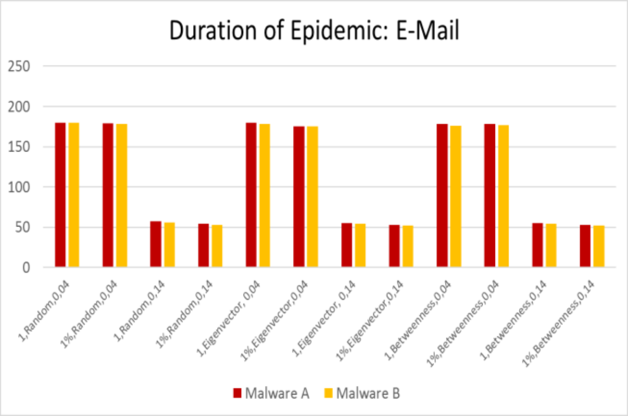


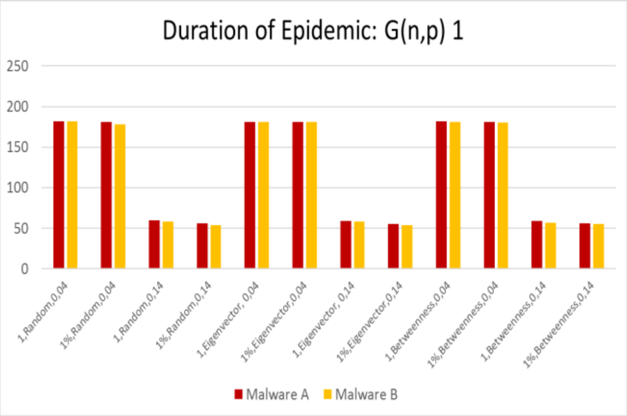


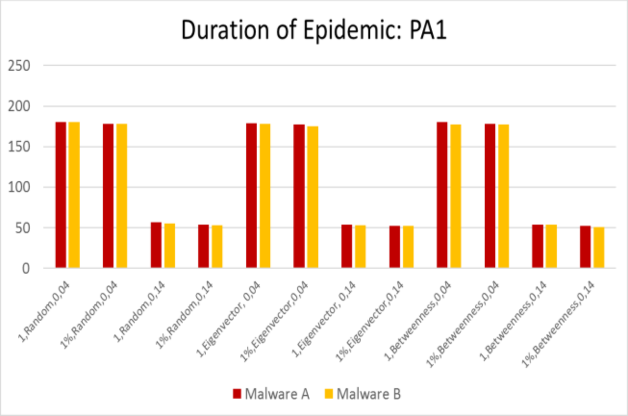


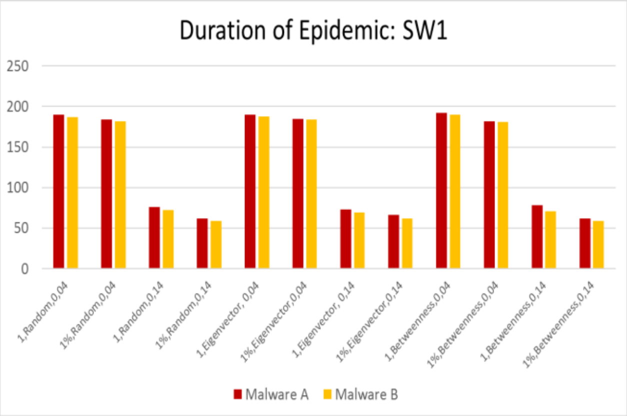
APPENDIX D

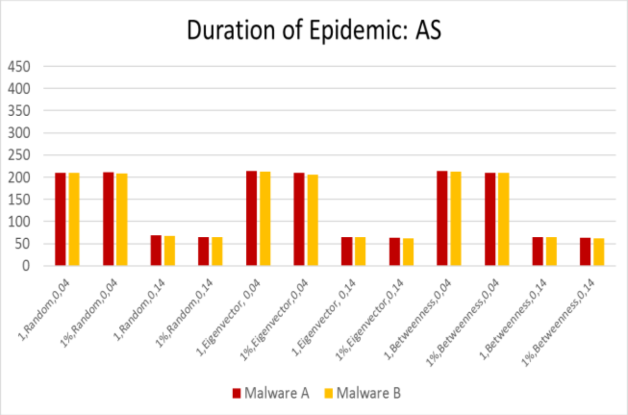
Prevalence – Duration of the Epidemic

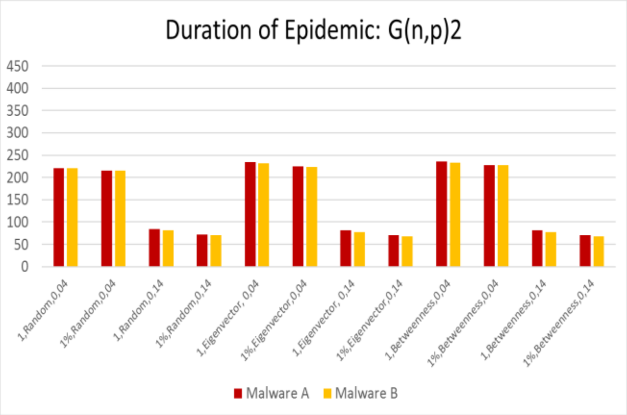


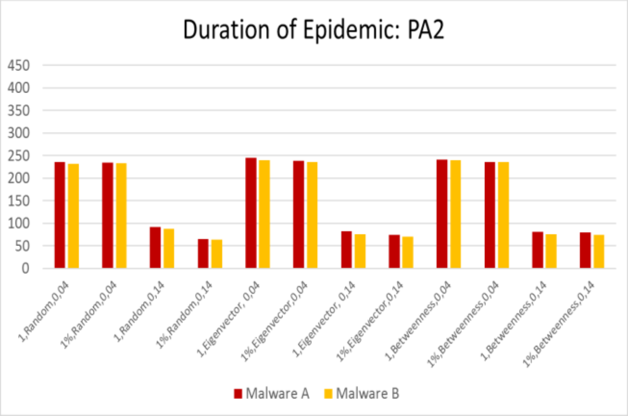


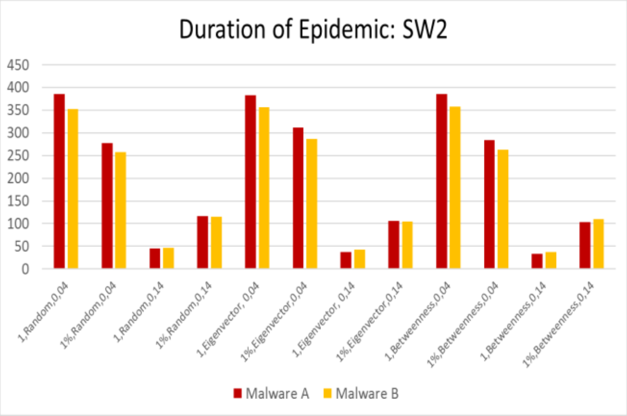






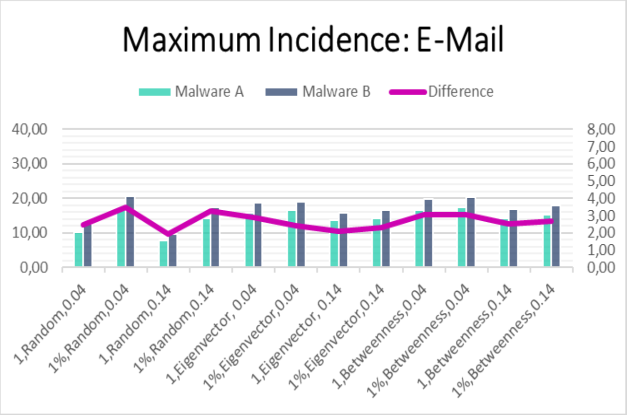


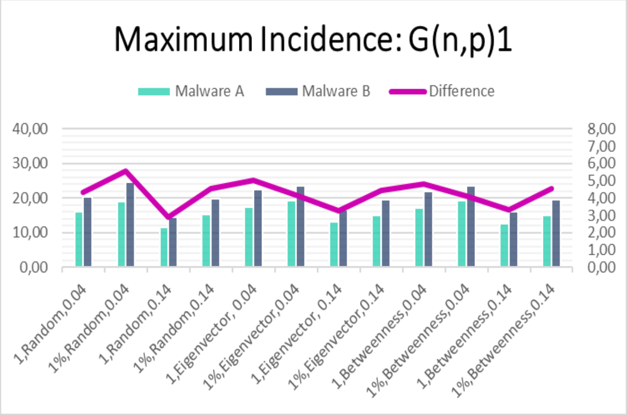


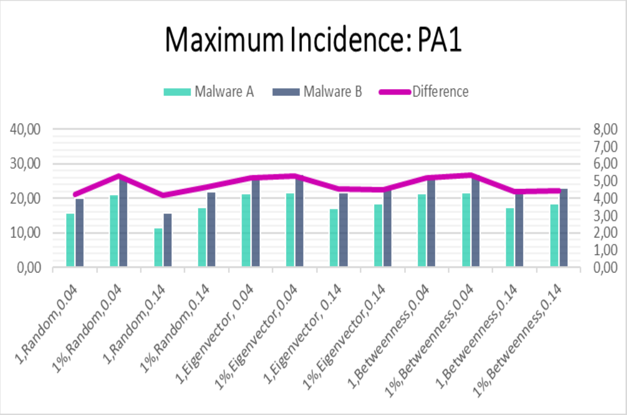


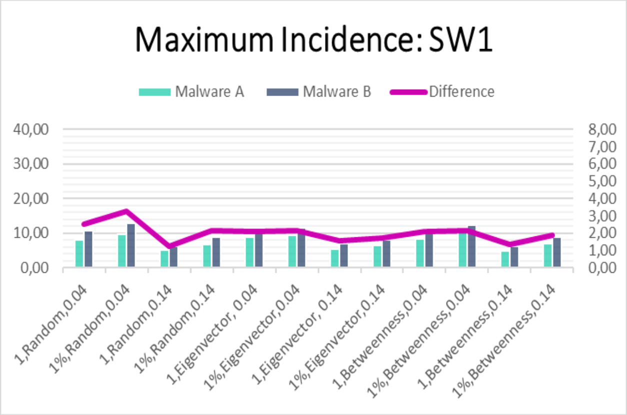
APPENDIX E

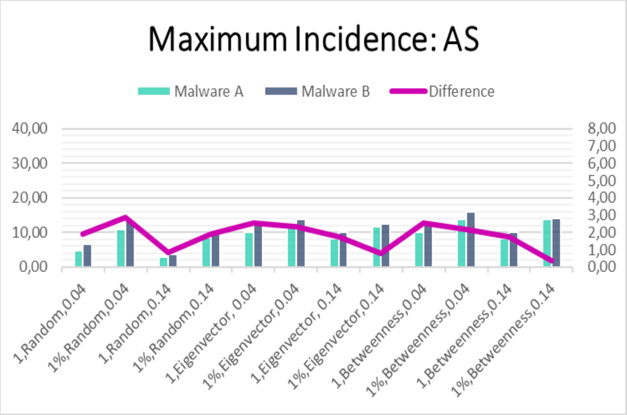
Incidence – Maximum Incidence (in percent)

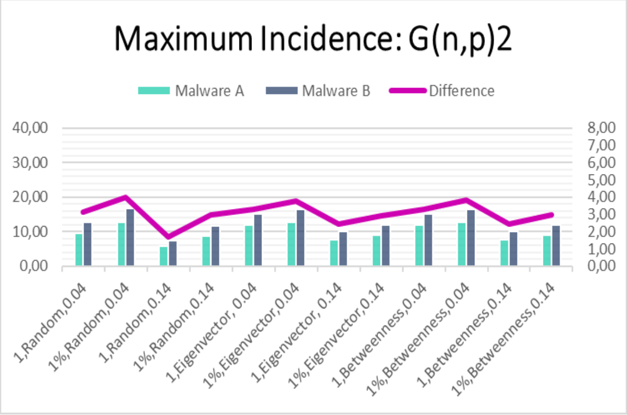


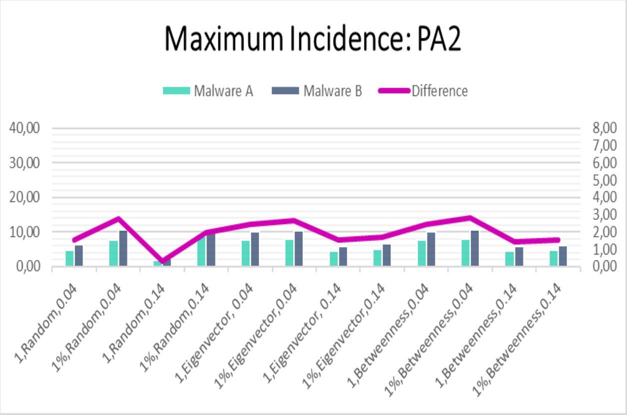


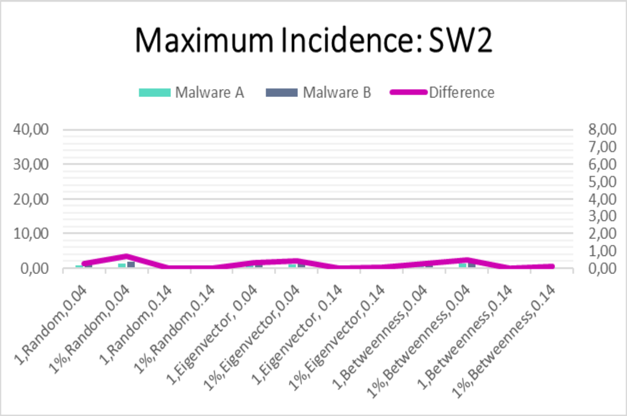






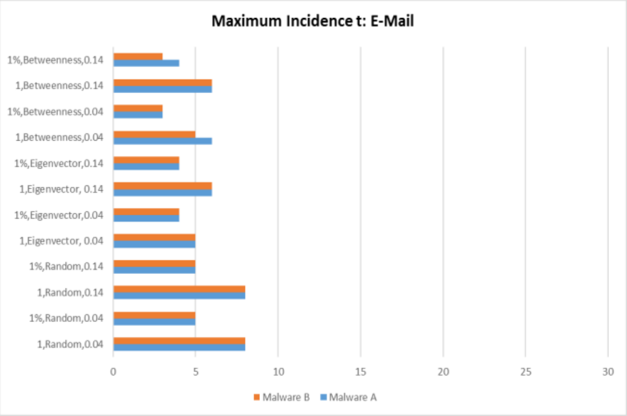


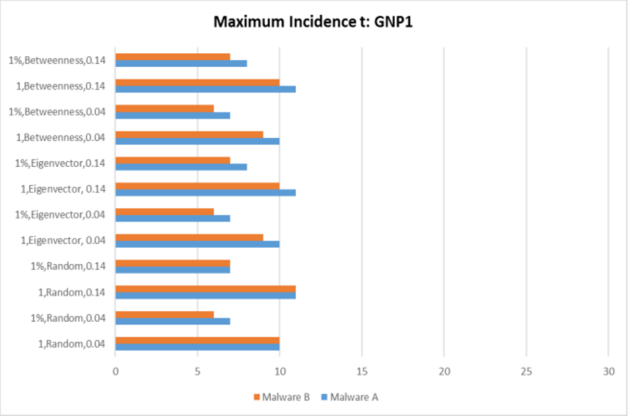


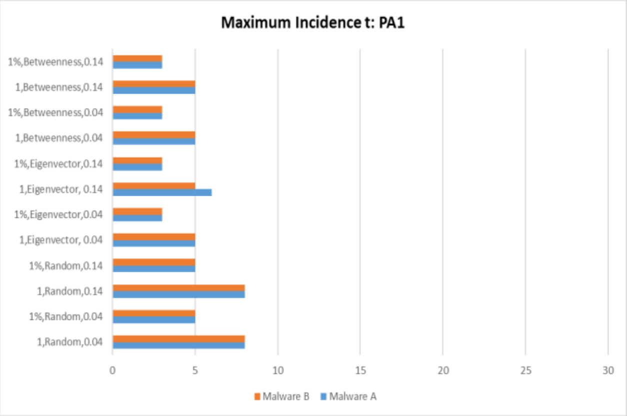


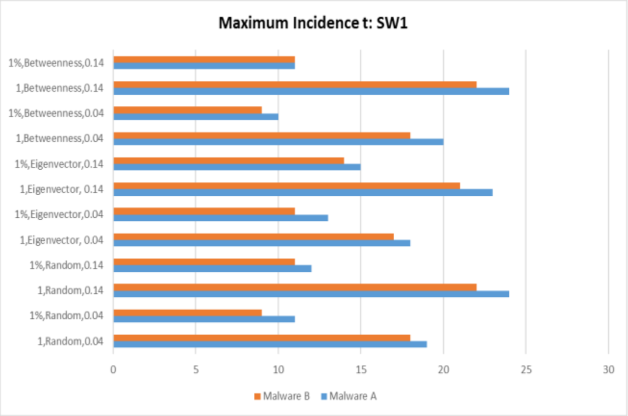
APPENDIX F

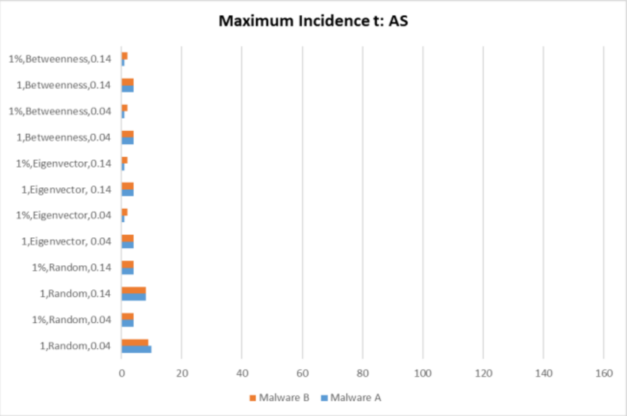
Incidence – Time until Maximum Incidence is reached

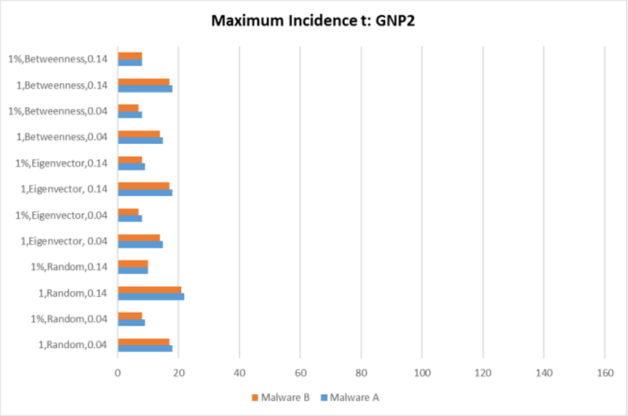


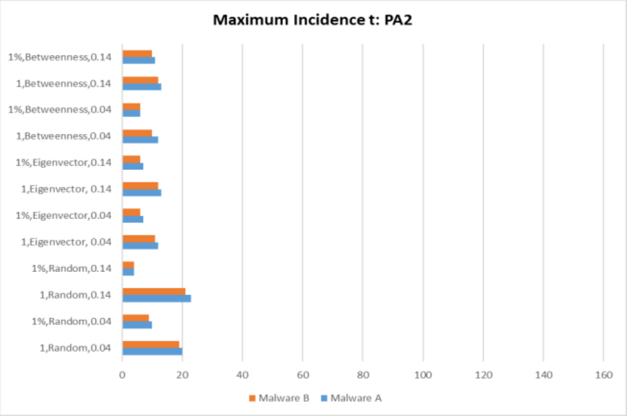


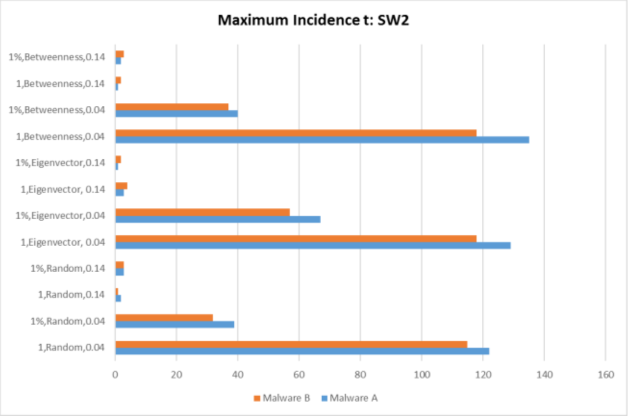






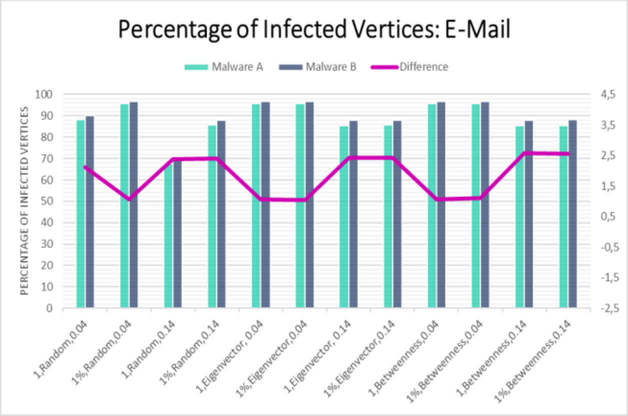


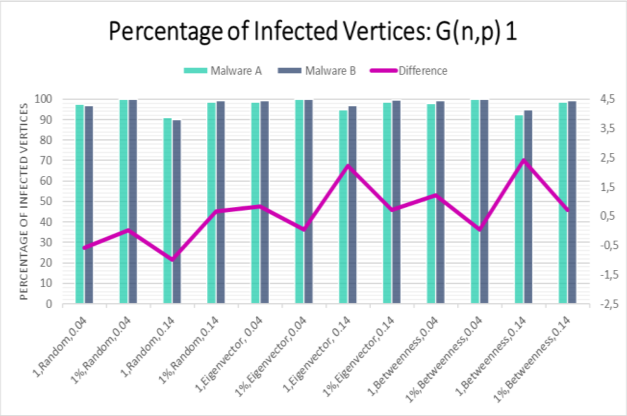


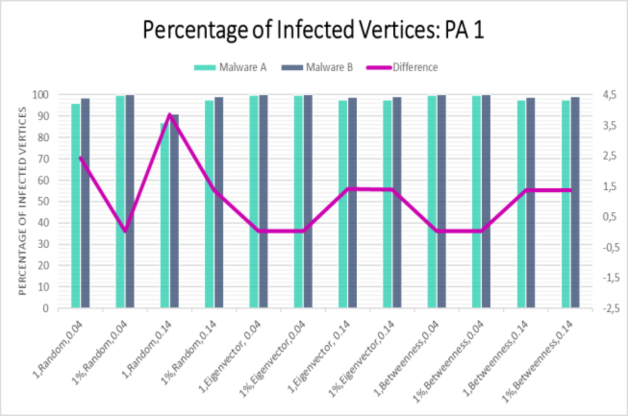


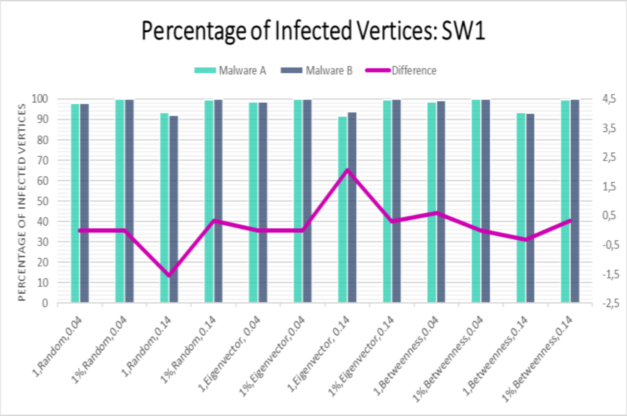
APPENDIX G

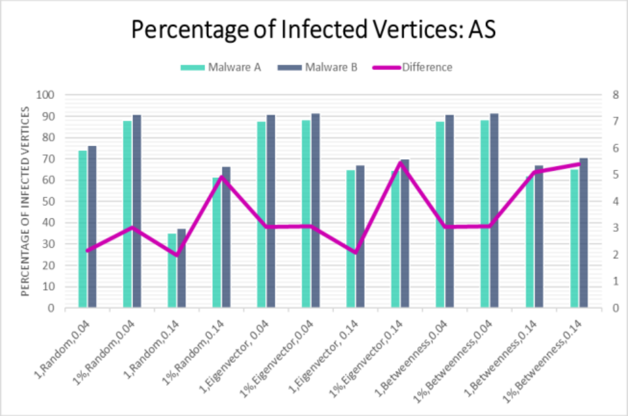
Percentage of Infected Vertices

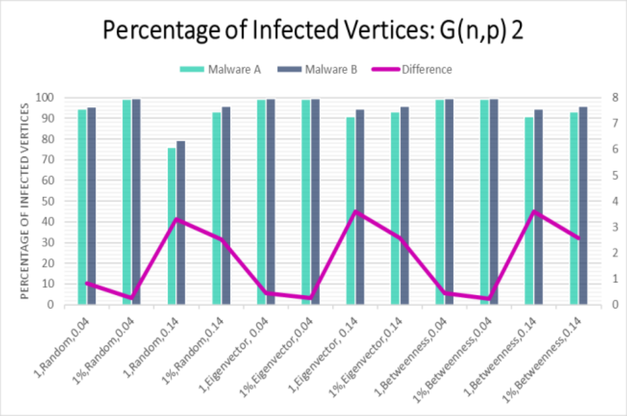

Percentage of Infected Vertices: E-Mail
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14

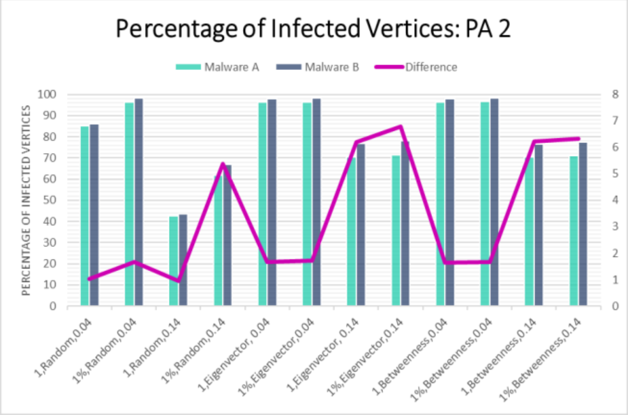

Percentage of Infected Vertices: G(n,p) 1
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14

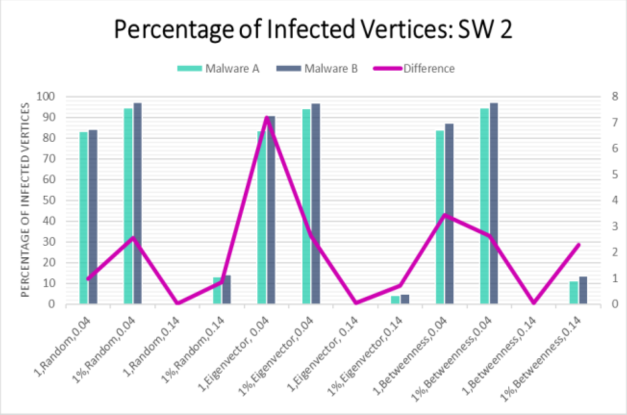

Percentage of Infected Vertices: PA 1
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14


Percentage of Infected Vertices: SW1
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14


Percentage of Infected Vertices: AS
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14


Percentage of Infected Vertices: G(n,p) 2
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14


Percentage of Infected Vertices: PA 2
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14


Percentage of Infected Vertices: SW 2
Malware A
Malware B
Difference
PERCENTAGE OF INFECTED VERTICES
1,Random,0.04
1%,Random,0.04
1,Random,0.14
1%,Random,0.14
1,Eigenvector, 0.04
1%,Eigenvector,0.04
1,Eigenvector, 0.14
1%,Eigenvector,0.14
1,Betweenness,0.04
1%,Betweenness,0.04
1,Betweenness,0.14
1%,Betweenness,0.14