

# Flip the virus: Modelling targeted attacks using Flipt with propagation delay

Sophie Marien

Thesis voorgedragen tot het behalen  
van de graad van Master of Science  
in de ingenieurswetenschappen:  
computerwetenschappen,  
hoofdspecialisatie Veilige software

**Promotor:**

Prof. dr. T. Holvoet

**Assessoren:**

Prof. dr. B. Jacobs

Dr. ir. A. Dries

**Begeleiders:**

Ir. Jonathan Merlevede,  
Ir. Kristof Coninx

© Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail [info@cs.kuleuven.be](mailto:info@cs.kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Many people have helped in the realization of this thesis. I would like to thank everybody who kept me busy the last year and made it possible for me to finish my thesis.

First I would like to thank my promoter and my assistants for their excellent support during the writing of this paper. They helped me figure out what to do and put a lot of time and effort in me.

Second I would like to thank my parents for their patient with me. A special thanks goes to my mother, who helped me a lot with my thesis and told me how to write in a structured way. Without her this thesis would be incomprehensible. Also a thank for my brother for his insights into mathematics that gave me a different angle to look at problems.

I would also like to thank my boyfriend who supported me during the writing of this thesis and helped me to write a beautiful text.

Last but not least all the people that read my thesis for faults: ..  
A few other people earn a special mention: Revue-blokt for keeping me focused on my thesis, DistriNet labo for letting me in and study in a quit place, Antonio for distracting me and telling me stories about his travels. A special memorial to John Nash, a mathematician with a fundamental contribution to Game Theory, that died in a car accident during the writing of this thesis. His Nash Equilibrium is used in this thesis..

*Sophie Marien*

# Inhoudsopgave

<b>Voorwoord</b>	<b>i</b>
<b>Samenvatting</b>	<b>iv</b>
<b>Samenvatting</b>	<b>v</b>
<b>Lijst van figuren en tabellen</b>	<b>ix</b>
<b>List of Abbreviations and Symbols</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
<b>2 Introduction to GameTheory</b>	<b>5</b>
2.1 What is cyber security? . . . . .	5
2.2 A brief introduction in Game Theory . . . . .	8
2.3 The FlipIt game . . . . .	10
2.4 Extensions on FlipIt . . . . .	13
<b>3 FlipIt game with virus propagation</b>	<b>17</b>
3.1 Introduction . . . . .	17
3.2 FlipIt game with virus propagation . . . . .	17
3.3 Explaining difference between FlipIt with and without virus propagation	18
3.4 Playing periodically with virus propagation . . . . .	19
<b>4 Nash Equilibria</b>	<b>27</b>
4.1 Nash Equilibria . . . . .	27
<b>5 Virus Propagation</b>	<b>35</b>
5.1 Methods of virus propagation . . . . .	35
5.2 REpresenting Virus Propagation in a network . . . . .	36
5.3 Related work . . . . .	36
5.4 Representing Virus Propagation in a network . . . . .	36
5.5 methode met matrixen . . . . .	36
<b>6 Related work</b>	<b>41</b>
6.1 Related work . . . . .	41
6.2 Conclusion . . . . .	42
6.3 Why Game Theory to model security problem . . . . .	42
6.4 .. . . .	42

<b>7 Conclusion</b>	<b>43</b>
7.1 trala . . . . .	43
<b>Bibliografie</b>	<b>45</b>

# Todo list

Results toevoegen . . . . .	iv
laatset zinnen niet correct . . . . .	vii
aanpassen naar chapter met nash equilibria . . . . .	3
what is cyber security? In deze tekst lijkt het op computer security . . . . .	6
opsplitsing uit het boek van network security genomen . . . . .	6
bekijk dit . . . . .	8
stukje geschiedenis toevoegen . . . . .	8
optimal function same as best response ? . . . . .	9
beter verwoorden . . . . .	14
dat laatste nog eens nakijken . . . . .	30

# Samenvatting

Recently, high profile targeted attacks such as the attack on Belgacom (a major Belgian telcom), have demonstrated that even the most secure companies can still be compromised, and that moreover such attacks can go undetected for a while. These kind of attacks are called APT, advanced persistent threats, which are designed to penetrate secretly a computer network, collect sensitive data and stay hidden for many years. A company has every interest to mitigate the risks of an APT and the consequences that it can cause. Because of the stealthiness fighting against these attacks requires methods that go beyond the standard tools against malware.

A group of researchers at the RSA, van Dijk et al., proposed the game FlipIt (The game of “stealthy takeover”) to model stealthy takeovers. It is a 2-players game composed of a single attacker, a single defender and a single shared resource. The players will compete to get control over the shared resource. Every move of the players will involve a cost and these moves happen in a stealthy way. The objective of the game for each player is to maximise the fraction of time of controlling the resource and minimise the total move cost.

FlipIt does however not take into account that a move may not be instantaneous, but has a certain delay. We adapt FlipIt such that we can use it to model the game of defending a company network that is attacked by a virus. The FlipIt formulas are adapted such as to take the delay for virus propagation into account, which in our case will be a delay for the attacker. In this paper, we restrict ourselves to games where both the defender and the attacker play with a periodic strategy. The goal of this thesis is to find out if there are interesting Nash Equilibria for a game with a virus propagation delay and if we can learn some lessons out of it.

**Keywords:** Game theory, Advanced persistent threats, cyber security, FlipIt, stealthy takeovers.

Results toevoegen

# Samenvatting

Onlangs zijn er gerichte security aanvallen geweest op grote bedrijven, zoals de aanval op Belgacom (een grote Belgische telecom). Deze aanvallen hebben aangetoond dat zelfs de meest veilige bedrijven nog steeds gecompromitteerd kunnen worden, en dat bovendien dergelijke aanvallen onopgemerkt kunnen blijven voor een bepaalde tijd. FlipIt is een model dat door onderzoekers van de RSA (van Dijk et al.) is voorgesteld om dergelijke aanvallen te modelleren. Het is een 2-spelers spel bestaande uit een aanvaller, een verdediger en een gedeelde bron. De spelers proberen om controle te krijgen over de gedeelde bron en ze doen dit op een heimelijke manier. Met FlipIt wordt echter geen rekening mee gehouden dat een aanval niet onmiddellijk is, maar dat dit kan gebeuren met een zekere vertraging. In dit artikel passen we het model van FlipIt zodanig aan dat we het kunnen gebruiken voor heimelijke aanvallen die onderheven zijn aan een vertraging. Resultaat :

FlipIt is een spel geïntroduceerd door Van Dijk et al. Om te begrijpen hoe we het FlipIt spel kunnen aanpassen om virus propagatie in acht te nemen, is het belangrijk om vertrouwd te raken met de concepten van het basische FlipIt spel en de notaties. Daarom beginnen we eerst met een uitleg hoe het basische FlipIt spel werkt en de belangrijkste formules die we in de paper gebruiken.

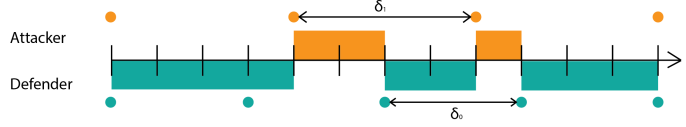
FlipIt is een spel met twee spelers met een gedeelde bron die de spelers zo lang mogelijk willen beheeren. De gedeelde bron kan een wachtwoord, een netwerk of een geheime sleutel zijn afhankelijk van welke situatie gemodelleerd wordt. In de rest van de paper noemen we de twee spelers de aanvaller, aangegeven met onderschrift  $A$  en de verdediger, aangegeven met onderschrift  $D$ .

Het spel begint op tijdstip  $t = 0$  en blijft voor onbepaalde tijd doorgaan ( $t \rightarrow \infty$ ). De tijd van het spel wordt aangenomen als continu. Om controle over de bron te krijgen kunnen de spelers  $i$ , met  $i \in \{A, D\}$  de bron flippen. Elke flip impliceert een zekere kost  $k_i$  en deze kosten kunnen variëren voor elke speler. Beide spelers proberen om hun kosten te minimaliseren. Door een kost in te voeren, voorkomt men dat de spelers te vaak bewegen.

De unieke eigenschap van FlipIt is dat elke flip gebeurt op een heimelijke manier, wat betekent dat de speler heeft geen idee dat de andere speler (zijn tegenstander) de bron heeft geflipt. Zo zal de verdediger niet kunnen achterhalen of de bron al is geflipt door de aanvaller tot hij de bron zelf flipt. Het doel van de speler is om zo lang mogelijk de controle te behouden over de bron en terwijl de kost minimaliseren.



Een beweging kan ook leiden tot een "verloren beweging", genoemd een flop. Het kan gebeuren dat de bron reeds onder controle is van de speler. Als de speler flipt wanneer hij of zij al de controle heeft over de bron, dan verspilt de speler een zet omdat het niet leidt tot een verandering van controle en dus een kost verspilt wordt.



FIGUUR 1: Een afbeelding van een Flipit spel met discrete tijdsintervallen waarbij beide spelers periodiek spelen. Elke beweging of flip wordt aangegeven met een blauw of oranje cirkel. De aanvaller is vertegenwoordigd in het oranje en speelt met een periode van  $\delta_A = 4$ . De verdediger is vertegenwoordigd in het blauw en speelt met een periode van  $\delta_D = 3$ . De blauwe en oranje rechthoeken geven de hoeveelheid tijd die de betreffende speler in de controle is van de bron.

De staat van de bron wordt aangeduid als een tijdsafhankelijke variabele  $C = C_i(t)$ .  $C_D(t)$  is 1 als het spel onder controle is van de verdediger en 0 als het spel onder controle is van de aanvaller. Omgekeerd, zal  $C_A(t)$  1 zijn als het spel onder controle is van de aanvaller en 0 als onder controle van de verdediger. Dus,  $C_A(t) = 1 - C_D(t)$ . Het spel begint met de verdediger in controle:  $C_D(0) = 1$ .

De spelers krijgen een benefit gelijk aan de hoeveelheid tijd dat ze in het bezit zijn van de bron min de kosten voor het maken van de bewegingen. De kosten van een speler  $i$  worden aangegeven met  $k_i$ . De totale winst van de speler  $i$  is gelijk aan de totale hoeveelheid tijd waarbij een speler  $i$  in controle is van de bron vanaf het begin van het spel tot de huidige tijd  $t$ . Dit wordt als volgt uitgedrukt:

$$G_i(t) = \int_0^t C_i(x) dx. \quad (1)$$

De totale winst van de verdediger opgeteld bij de totale winst van de aanvaller telt op tot  $t$ :

$$G_D(t) + G_A(t) = t \quad (2)$$

De gemiddelde winst van speler  $i$  wordt gedefinieerd als volgt:

$$\gamma_i(t) = G_i(t)/t. \quad (3)$$

En daarmee voor alle  $t > 0$ :

$$\gamma_D(t) + \gamma_A(t) = 1 \quad (4)$$

Laat  $\beta_i(t)$  is de gemiddelde benefit van een speler  $i$  tot aan tijd  $t$ :

$$\beta_i(t) = \gamma_i(t) - k_i \alpha_i. \quad (5)$$

Dit is gelijk aan de fractie van de tijd waarbij de bron in handen is van speler  $i$ , minus de kosten van het maken van de bewegingen.  $\alpha_i$  definieert het gemiddeld aantal flippen door speler  $i$  tot tijd  $t$ . De asymptotische benefit ratio wordt gedefinieerd als  $\liminf$  van de gemiddelde benefit omdat de tijd  $t$  toeneemt tot oneindig en de gemiddelde benefit niet beperkend waarden.

$$\beta_i(t) = \lim_{t \rightarrow \infty} \inf \beta_i(t) \quad (6)$$

laatset zinnen  
niet correct

## strategieën

Omdat de spelers bewegen in een sluipende manier, zijn er verschillende soorten feedback die een speler kan krijgen tijdens het rijden. Dergelijke feedback kan worden verdeeld in twee groepen van strategieën. De niet-adaptieve strategieën en de adaptieve strategieën. Deze worden beschreven in de tabel 1.

Als er geen feedback voor geen van beide spelers, hebben wij een niet-adaptieve strategie. Omdat een speler geen feedback tijdens het spel zal hij spelen op dezelfde manier tegen elke tegenstander. De strategie die niet-adaptieve omdat de speelstrategie niet afhankelijk is van de tegenstander bewegingen. Een interessante subklasse van de niet-adaptieve strategieën die waar de tijdsintervallen tussen twee opeenvolgende bewegingen worden gegenereerd door een vernieuwingsproces. Een voorbeeld hiervan werd strategie is de periodieke strategie waarbij het tijdsverloop tussen twee opeenvolgende bewegingen van de spelers een vast interval. Een exponentiële strategie is vernieuwingsstrategie waarbij het interval tussen twee opeenvolgende zetten exponentieel verdeeld.

In het geval dat er feedback, een speler kan zijn strategie aanpassen aan de ontvangen over de bewegingen van de tegenstander informatie. Afhankelijk van de hoeveelheid informatie ontvangen, kunnen twee subklassen van adaptieve strategieën worden geïdentificeerd. The Last Move (LM) strategieën vertegenwoordigen de klasse waarin wanneer een speler flips zal hij vinden van de exacte tijd dat de tegenstander speelde de laatste keer. In de tweede klasse, genaamd Full History (FH), wanneer een speler flips zal hij vinden van de hele geschiedenis van de beweging van de tegenstander. In dit artikel zullen we ons richten op de niet-adaptieve strategieën. Deze keuze is ingegeven door het feit dat in een zekerheid spel een speler (verdediger of aanvaller) zelden informatie over de bewegingen (laatste zet of volledige geschiedenis) van zijn tegenstander.

Het onderzoek van de verschillende strategieën middels Flipit framework stelt een aantal interessante resultaten leiden:

- periodieke spellen domineren de andere vernieuwing strategieën, wat betekent dat het altijd voordelig om periodiek tegen een tegenstander met een vernieuwing van de strategie te spelen;

Categories	Klassen Strategies
Non-adaptieve (NA)	Vernieuwing - Periodieke - Exponentiële General non-adaptieve
Adaptive (AD)	Laatste move (LM) Full History (FH)

TABEL 1: hiërarchie van de klassen van de strategieën in Flipit

- periodieke games zijn nadelig tegen spelers na een laatste zet adaptieve strategie;
- als de verdediger speelt met een periodieke tarief dat snel genoeg is zal hij de aanvaller te dwingen uit te vallen;
- 

In dit *abstract* environment wordt een al dan niet uitgebreide Nederlandse samenvatting van het werk gegeven. Wanneer de tekst voor een Nederlandstalige master in het Engels wordt geschreven, wordt hier normaal een uitgebreide samenvatting verwacht, bijvoorbeeld een tiental bladzijden.

Dit is het kort geschreven.

Eigenlijk paper in het nederlands schrijven

Intro – positionering – vraagstelling – oplossingen en contributies

Flipit en cybersecurity

Flipit met delay

Berekeningen

Niet zo lang geleden werden APT's ontdekt. (Introzin moet beter en anders). Bedrijven worden door APTs aangevallen. Belangrijk om APTs tegen te gaan. Geven een grote kost aan de bedrijven. Hun stealthy aspect is vervelend en moeilijk aan te pakken. Preventie is beter dan detectie. Eens ze binnen zijn kunnen we veel schade veroorzaken. Conventionele middelen zoals firewalls, malware detection helpen hier niet tegen. Nieuwe methode nodig. Gametheory FlipIt. FlipIt uitleggen. Deze formules gaan we dan omvormen tot ze wel propagatie delay kunnen modelleren. Hiervan berekenen we ook het Nash equilibrium.

# Lijst van figuren en tabellen

## Lijst van figuren

- 1 Een afbeelding van een FlipIt spel met discrete tijdsintervallen waarbij beide spelers periodiek spelen. Elke beweging of flip wordt aangegeven met een blauw of oranje cirkel. De aanvaller is vertegenwoordigd in het oranje en speelt met een periode van  $\delta_A = 4$ . De verdediger is vertegenwoordigd in het blauw en speelt met een periode van  $\delta_D = 3$ . De blauwe en oranje rechthoeken geven de hoeveelheid tijd die de betreffende speler in de controle is van de bron. . . . . vi
- 2.1 Prisoners dilemma: An example from the field of Game Theory. It is a game between two perfectly rational prisoners who don not know what the other one will do. Each can confess or stay silent. The outcomes can be seen in the picture. Prisoner 1 corresponds to each first column, Prisoner 2 corresponds to each second column. . . . . 10
- 2.2 A representation of a FlipIt game where both players are playing periodically. Every move or flip is indicated by a blue or orange circle. The attacker is represented in orange and plays with a period of  $\delta_A = 4$ . The defender is represented in blue and plays with a period of  $\delta_D = 3$ . The blue and orange rectangles represent the amount of time the respective player is in control of the resource. . . . . 11
- 3.1 The first game is the basic FlipIt game. The second is a FlipIt game with a delay. During the first flip of the attacker, the defender moves after the delay, causing the attacker to get control over the resource. During the second flip of the attacker, the defender flips at time  $t+d$ , causing the defender to take control over the resource before the attacker. During the third and final flip of the attacker, the defender flips during time  $t+d$ , causing the attacker to never gain control over the resource. 18

3.2	Formalization of a FlipIt game with delay: A representation of a FlipIt game where both players are playing periodically. Every move or flip is indicated by a blue or orange circle. The defender is represented in blue and plays with a period of $\delta_D$ . The flip of the attacker is represented by a white circle, but because there is a delay $d$ , the attacker only controls the resource after time $d$ represented by an orange circle. The attacker plays with a period of $\delta_A$ . The blue and orange rectangles represent the amount of time the respective player is in control of the resource. . . . .	20
3.3	Case 1: Difference between a basic FlipIt game and a FlipIt game with a delay. (1) is the FlipIt game without a delay and (2) is with a delay. The delay is denoted with an arrow. The attacker is only in control when the circle becomes orange. . . . .	21
3.4	All possible cases for the attacker and the defender in Case 2.A where $d + \delta_A < \delta_D$ . As can be seen, the defender will have control during a period of $d$ over the resource when the defender has flipped in the previous interval. . . . .	24
4.1	This figure shows the three subcategories of each case. 'A' stands for Case 2.A: $\delta_D \geq d + \delta_A \geq \delta_A$ and 'B' stands for Case 2.B: $d + \delta_A \geq \delta_D \geq \delta_A$	28

## Lijst van tabellen

1	hiërarchie van de klassen van de strategieën in Flipit . . . . .	viii
2.1	Hierarchy of Classes of strategies in FlipIt . . . . .	13

# List of Abbreviations and Symbols

## Abbreviations

APT	Advanced Persistent Threat
SIR	Susceptible - Infected - Recovered
SIS	Susceptible - Infected - Susceptible

## Symbols

$i$	Defines the player. Different as in the FlipIt paper where the defender is denoted by the subscript 0 and the attacker by the subscript 1.
$\delta_i$	The length of the interval between two consecutive moves of player $i$ .
$\alpha_i$	The average flip rate of player $i$ , given by $\alpha_i = 1/\delta_i$ .
$k_i$	The cost of player $i$ 's moves.
$d$	The delay caused by the virus propagation.
$G_i(t)$	The total gain of player $i$ denotes the amount of time player $i$ is in control over the resource up to time $t$ .
$\gamma_i$	The average gain rate of player $i$ defined as $G_i(t)/t$ .
$\beta_i$	The average benefit rate up to time $t$ defined as $\beta_i = \gamma_i - k_i\alpha_i$ .
$opt_i$	The optimum function.

# Hoofdstuk 1

## Introduction

### 1.1 Introduction

#### Situation

In this era where digitalization becomes prominent in every aspect of our lives, where technology is growing fast and where businesses are always under attack, security becomes an issue of increasing complexity. Without security, there is no protection to keep somebody out of a system. It is the same as leaving the door of your house wide open for everyone to come in.

Why is it so important to keep a system secure? Many businesses store confidential information on clients, which can be lost and possibly be abused by competitors through data leakage. Also, disruption caused by distributed DoS attacks, may result in businesses failing to meet their service-level agreements. Ultimately, system and network security helps protecting a business's reputation, which is one of its most important assets.

A particular kind of threats are Advanced Persistent Threats (APT). An APT is a multi-faceted, continuous and targeted cyber attack that is designed to penetrate a network or a system in a stealthy way and that can stay undetected for a long period. It is something different and stronger than a conventional threat. A conventional threat will not attack any particular target. An APT is persistent and will keep on trying to attack its victim. It operates silently and stealthily, to prevent detection. This makes it so hard to protect a network or a system against an APT.

#### Complication

Since it is so difficult to protect a system or a network against APT's, researchers have been looking for effective ways to predict in advance which defence strategy might be the better one. Game theory is gaining increasing interest as an effective technique to model and study cyber security problems. It is common to model cyber security games as a game with two players, an attacker and a defender. However,

there are games that have more players [3]. In this paper we focus on the game with two players. The actions available to the attacker and the defender correspond to attacks on the system and defensive measures that protect the system. In particular, both players will execute the strategy that results in a maximal benefit for them. Researchers at RSA, van Dijk et al, made a game theoretic model of targeted attacks. They study the specific scenario where a system or network is repeatedly taken over completely by an without being immediately detected by the defender of the system or network. In game theory, such a game is known as “FlipIt” [14]. This is a two players game where the attacker and the defender are competing to get control over a shared resource. Neither player knows who is currently in control of the resource until they move. In FlipIt every move gives them immediate control over the resource. But what if the attacker moves and it takes a while before the attacker gets full control over the resource? FlipIt does not take into account that a move may not be instantaneous, but has a certain delay. Consider for example a network with different nodes ( laptops, datacenters) as a resource. The attacker drops a virus on one of the nodes and then wait till this virus infects the whole network. The attacker will only be in control of the resource once the whole network is infected.

### Research questions

This paper adapts the model presented in [14] so as to take the delay for virus propagation into account. This leads us to the following research questions:

- How can we incorporate the notion of delay in the game-theoretical analysis of the Flip-It game for a periodic strategy ?
- Is there an optimal defense strategy against an attacker ?

### Contributions

We propose an addition to the basic FlipIt model to model a scenario where the moves by the attacker will not be instantaneous. Next we analyse what the new Nash equilibria will be and ..

### Overview of the thesis

The organisation of this paper is the following. In chapter 2 a brief introduction to cyber security and gametheory is introduced to get familiar with the kind of threats that are in the scope of this work and the game theoretic concepts that will be further used in the paper. In the same chapter the FlipIt framework is summarized and the most important conclusions together with the the related work done on FlipIt and the difference with this paper. In chapter 3 , we first introduce the adaptations made on FlipIt to model a FlipIt game with virus propagation. After that formulas are derived to model a FlipIt game with a virus propagation for a specific case where



players play a periodic strategy with a random phase.

In Chapter 4 the formulae are used to determine if there is a Nash equilibrium. Next in Chapter 5 we give an overview of the various ways in which a virus can propagate. We present a method to calculate the speed of the propagation of a virus in a network and how this network can be established to reduce the spreading of a virus. Finally, in Chapter 7, we discuss the main results and complications and provide directions for further research.

aanpassen naar  
chapter met  
Nash equilibria



## Hoofdstuk 2

# Introduction to GameTheory

This chapter provides the reader with an introduction to the general context of the work presented in this paper. Section 2.1 introduces the reader into the basic concepts of cyber security and the kind of cyber security threats that are in the scope for this work. Section 2.2 then introduces the reader into the main principles of game theory. Subsequently, section 2.3 introduces the reader to the FlipIt game: the game that will be used to model cyber security attacks of a periodic nature, including a delay. Finally, section 2.4 gives an overview of the related work, and how the research presented in this paper compares to existing results.

### 2.1 What is cyber security?

Before the digitalization of documents, information was kept on paper and the security of this information was ensured by administrative and physical means. For example, you needed a key to access documents stored in a room full of cabinets where the files were kept. In today's digital era more and more information is kept in a digital format, stored on a computer. As digitalization progressed, the need for ensuring the security of digital information arose and automated tools were developed to protect files stored on a computer.

The generic term for data protection stored on a computer controlled device such as computers and smartphones, as well as public and private computer networks, including the entire Internet is called computer security. Security is a general term that encompasses several dimensions. More specifically, computer security has three key objectives that are fundamental to computer security:

1. *Confidentiality*: assuring that the confidentiality of private data is not disclosed or made available to users that do not have proper authorization.
2. *Integrity*: assuring that data cannot be altered by an unauthorized individual.
3. *Availability*: assuring that data is always accessible and that the service is not denied to authorized individuals.

These key attributes are also known as the CIA triad. They are the fundamental security objectives for securing data, information and computing services.

A subset of computer security is cyber security. This is the process of applying security tools to ensure confidentiality, integrity, and availability of data.

what is cyber security? In deze tekst lijkt het op computer security

It is an attempt to protect websites, servers, data, applications, operating systems and all assets that need protection in a computer system. Some of these tools may include detection, identification or removal tools. A detection tool will determine if an infection has taken place and will trace the threat. An identification tool will try to identify the threat to be able to know how to remove it. The removal tool will remove the threat from the system (once it has been identified) so that it cannot spread any further.

### Threats to computer systems

In order to keep a system secure and to be able to apply security tools, it is important to mitigate the possible threats. The possible threats can take many forms, the most common being: spam, malware, spoofing, phishing and DDoS attacks. In the context of cyber security, the terms ‘threat’ and ‘attack’ are often used interchangeably, referring to more or less the same thing. The meaning of these two terms, however, differ slightly from one another. The former refers to anything that can breach security and cause possible harm. It is a possible danger that can exploit a vulnerability. The latter is an assault on computer security that originates from a threat. It is an intelligent act that deliberately tries to breach security through vulnerabilities.

The most noteworthy and biggest group of threats to computer systems is malware. This is a piece of malicious software that is designed to penetrate unprotected systems or computers, with the intent to retrieve sensitive information, destroy data, or compromise the confidentiality, integrity or availability of the data or applications of the victim. A security report of 2014 [4] reveals that 61% of the attacks on companies are caused by malware. For this reason this section will examine the categories of malware threats. Different types of malware exist: virusses, worms, flooders, rootkits, bots, spyware, adware and many more. This broad range of different types can be classified into two main categories, the first one based on the propagation method that is used and the second one for the payload or the variety of actions that the malware performs. Propagation methods include virusses, worms and trojans. Payload includes e.g. flooders, rootkits, bots, spyware and adware. This paper focuses on the category of propagation methods and not on the actions that the malware performs. A brief explanation of the three main propagation methods of malware is given below.

opsplitsing uit het boek van network security genomen

*Virus:* This is a malicious piece of code that replicates itself and tries to spread in order to infect other systems or files. A typical virus will attach itself to a program, or an executable content on a computer. The “I love you” virus is an

example of a virus where the virus attached itself as an executable to a mail. It propagated by using the mailing systems. When a victim opened an email with the “I love you” virus in the annex, the virus spread itself by sending a mail to everyone in the victim’s contact list.

Using this method, a virus can multiply rapidly, possibly even causing a business network to shut down by the heavy traffic. A virus needs human interaction to spread. In the example above: if no one were to open the mail, the virus could not spread itself and infect other systems.

*Worm:* A worm is a virus that can spread without human interaction. It is a computer program that replicates itself in order to spread to other hosts on a network. Copies of the worm can be forwarded via a computer network without an intermediary. The worm will use vulnerabilities of the system to infect other computers. The Stuxnetworm is a very prominent example of a worm. Initially this worm was spread via infected USB sticks and from there on it could spread itself to other hosts on the network through the Internet, without any further human intervention. The purpose of the Stuxnetworm was to harm the centrifuges in nuclear reactors; many reactors have been infected.

*Trojan:* This is a malicious program that disguises itself as something normal and useful, so that users won’t be suspicious of installing it, but it has a malicious function hidden inside that can avoid security measures and cause harm. A notable trojan horse is Koobface, that targeted users of facebook, skype, yahoo, gmail and AOL mail. To spread itself the worm sent a mail or friend request with a message that directed the recipients to a third party website. This site would then convince the recipient into downloading an update of Adobe Flash Player. Once downloaded and executed, Koobface could infect the host.

The threats caused by the malware described above can be divided into three main categories: known threats (70%), unknown threats (29%) and advanced threats (1%) as proposed by Kasperksy [5].

The known threats are the easiest to defend oneself against. Standard malware protection tools like firewalls and virus scanners can keep these kind of malware out of the system. Installing protection against unknown threats is also relatively easy, but this requires tools that go beyond the standard methods, e.g. dynamic whitelisting. The remaining 1% are the advanced threats, also known as APT. They are the most difficult to deal with.

### **Advanced threats to computer systems**

An Advanced Persistent Threat (referred to in the rest of this work as APT) is a persistent targeted attack that tries to penetrate a network to cause harm while staying unseen for a long period of time. The motive of an APT is mostly cyber espionage, stealing sensitive data, sabotage or other kinds of ideological attacks.

APT's are 'advanced' because these attacks are well funded and (usually) the attacker needs a great amount of expertise to successfully penetrate a network. Not all APT's are *technically* advanced though. The attacker can also try to exploit known vulnerabilities, in the hope that his target has not yet secured himself against them. 'Persistent' refers to the fact that the attacker never quits trying to attack his target. The attack can be spread over several years, taking multiple steps. An APT can be a mix of different types of malware and may use various propagation methods.

According to a security survey of Kaspersky [4] the damage of one successful targeted attack against a large company can exceed over 2.54 million dollar. Companies need a defence mechanism to defend themselves against APT's. As previously stated, simple detection and identification tools are insufficient to protect oneself against APT's, and a removal tool will only work if the threat has been identified. To mitigate these kind of attacks another security countermeasure is needed. This is where the researchers of RSA come into the picture. They have modelled these targeted attacks using Game Theory, and - given a certain amount of information - determined optimal defence strategies.

bekijk dit

### 2.2 A brief introduction in Game Theory

Gametheory is a mathematical study to analyse interactions between independent and self-interested agents. To get an understanding of the most important concepts of game theory, a short introduction based on the work of [9] and [10] is given in this section. For a more detailed and full introduction to game theory, the reader is referred to [9].

stukje geschiedenis toevoegen

Game theory studies the interaction between independent and self-interested agents. It is a mathematical way of modelling the interactions between two or more agents where the outcomes depend on what everybody does and how it should be structured to lead to good outcomes. It has therefore important applications in many area's such as economics, politics, biology, computer science, philosophy and a variety of other disciplines.

One of the assumptions underlying game theory is that the players of the game, the agents, are independent and self-interested. This does not necessarily mean that they want to harm other agents or that they only care about themselves. Instead it means that each agent has preferences about the states of the world he likes. These preferences are mapped to natural numbers and are called the utility function. The numbers are interpreted as a mathematical measure that tells how much an agent likes or dislikes the states of the world.

In a Decision Game Theoretic Approach an agent will try to act in such a way to maximise his expected or average utility function. It becomes more complicated

when two or more agents want to maximise their utility and when actions of the agents can affect each other's utilities. This kind of games are referred to as non-cooperative game theory, where the basic modelling unit is the group of agents. The individualistic approach, where the basic modelling is only one agent, is referred to as cooperative game theory.

### Best response and Nash Equilibrium

One of the solution concepts in Game Theory for non-cooperative games is a Nash Equilibrium that will be used in this paper. A Nash Equilibrium is a subset of outcomes that can be interesting to analyse a game. To define this concept we first introduce the concept of best response. The best response for a player is the action of a player that maximizes its pay-off for any given action of the other player. We define  $Opt_i$  as the best response function for player  $i$ . The best response for player 1 is given by :  $a_1 = Opt_1(a_2)$ . For a Nash Equilibrium each player has a consist list of actions and each player's action maximizes his or her pay-off given the actions of the other players. Nobody has the incentive to change his or her action if an equilibrium profile is played. We have a Nash Equilibrium for the pair  $(a_1^*, a_2^*)$  where  $a_1^* = Opt_1(a_2^*)$  and  $a_2^* = Opt_2(a_1^*)$

An example to explain a Nash equilibrium of a two-player game, for example, is the prison dilemma. In this game there are two players who are both rational and both of them have committed a crime. 'Rational' means that they want the best for themselves and it is not their purpose to do harm to others. Both are locked in a separate room and they cannot tell in advance to each other what they are going to say. Each of them can betray the other or they can support each other and remain silent. If a player confesses, he will, depending on what the other does, go for three years in prison or he is free to go. If the player is silent, he will, depending on what the other player does, go for five years in jail or only one year.



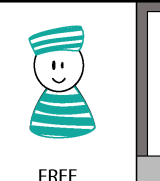





Figure 2.1 shows the rewards of the possible actions of the two players. Because betraying the other player gives a greater reward than remaining silent, all prisoners will betray each other which leads to the outcome that both of them will betray each other. This is because this is the dominant strategy. A dominant strategy is a strategy that is better than all other strategies of a player regardless of the opponent. Here it is advantageous for each player to confess. This gives them three years in prison instead of five year or they will be free instead of being in prison for one year. The Nash equilibrium of the game is that they both confess even though that it would be better if they both cooperated and chose to stay silent.

optimal function same as best response ?

### List of terms

In the following list a couple of terms that will be used throughout the paper.

FIGUUR 2.1: Prisoners dilemma: An example from the field of Game Theory. It is a game between two perfectly rational prisoners who don not know what the other one will do. Each can confess or stay silent. The outcomes can be seen in the picture. Prisoner 1 corresponds to each first column, Prisoner 2 corresponds to each second column.

		Prisoner 2			
		Talk		Stay Silent	
Prisoner 1	Talk	 3 YEARS	 3 YEARS	 FREE	 5 YEARS
	Stay Silent	 5 YEARS	 FREE	 1 YEAR	 1 YEAR

*Players:* Players are referred as the ones who are the decision makers. It can be a person, a company or an animal. (they will act rational )

*Actions:* Every player has actions that he or she can do.

*Strategies:* A strategy is the combination of different actions. A pure strategy is only one action.

*Utility:* The utility function or also known as the pay-of is the mapping of the level of happiness of an agent about the state of the world to natural numbers.

A game in game theory consists of multiple agents and every agent has a set of actions that he can play.

### 2.3 The FlipIt game

FlipIt is a game introduced by van Dijk et al. To understand how to model a FlipIt game with virus propagation it is important to get familiar with the concepts of the normal FlipIt game and its notations. Therefore, we first explain the framework of FlipIt and introduce the most important formulas that will be used throughout the paper.

FlipIt is a two-players game with a shared single resource that the players want to control as long as possible. The shared resource can be a password, a network or a secret key depending on the setting being modelled. In the remainder of the paper



we name the two players the attacker, denoted by the subscript  $A$  and the defender, denoted by subscript  $D$ .

The game begins at  $t = 0$  and continues indefinitely ( $t \rightarrow \infty$ ). The time in the game is assumed as being continuous. To get control over the resource, the players  $i$ , with  $i \in \{A, D\}$ , can flip the resource at any given time. Each move implies a certain cost  $k_i$  and can vary for each player. Both players try to minimize their cost. Adding a cost prevents players to move too frequently.

The unique feature of FlipIt is that every move happens in a stealthy way, meaning that the player has no clue that the other player (his adversary) has flipped the resource. For instance, the defender does not find out if the resource has been compromised by the attacker until he flips the resource himself. The goal of the player is to maximize the time that he or she has control over the resource while minimizing the total cost of the moves. A move can also result in a "wasted move", called a flop. It may happen that the resource was already under control by the player. If the player moves when he or she has already control over the resource, he or she would have wasted a move since it does not result in a change of ownership, so the cost is wasted.

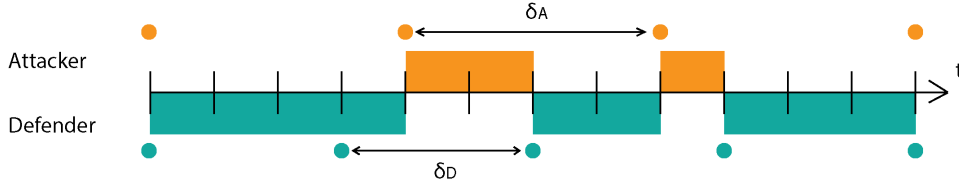


FIGURE 2.2: A representation of a FlipIt game where both players are playing periodically. Every move or flip is indicated by a blue or orange circle. The attacker is represented in orange and plays with a period of  $\delta_A = 4$ . The defender is represented in blue and plays with a period of  $\delta_D = 3$ . The blue and orange rectangles represent the amount of time the respective player is in control of the resource.

We denote the state of the resource as a time-dependent variable  $C = C_i(t)$ .  $C_D(t)$  is 1 if the game is under control by the defender and 0 if the game is under control by the attacker. Reversely,  $C_A(t)$  is 1 if the game is under control by the attacker and 0 if under control by the defender. So,  $C_A(t) = 1 - C_D(t)$ . The game starts with the defender being in control:  $C_D(0) = 1$ .

The players receive a benefit equal to the time units they were in possession of the resource minus the cost of making their moves. The cost of a player  $i$  is denoted by  $k_i$ . The total gain of player  $i$  is equal to the total amount of time that a player  $i$  has owned the resource from the beginning of the game up to time  $t$ . It is expressed as follows:

$$G_i(t) = \int_0^t C_i(x) dx. \quad (2.1)$$

If we add up the gain of the defender and the gain of the attacker it should sum up to  $t$ :

$$G_D(t) + G_A(t) = t \quad (2.2)$$

The average gain rate of player  $i$  is defined as:

$$\gamma_i(t) = G_i(t)/t. \quad (2.3)$$

And thus for all  $t > 0$  :

$$\gamma_D(t) + \gamma_A(t) = 1 \quad (2.4)$$

Let  $\beta_i(t)$  denote player's  $i$  average benefit upto time  $t$ :

$$\beta_i(t) = \gamma_i(t) - k_i \alpha_i. \quad (2.5)$$

This is equal to the fraction of time the resource has been owned by player  $i$ , minus the cost of making the moves.  $\alpha_i$  defines the average move rate by player  $i$  up to time  $t$ . In a given game, the asymptotic benefit rate (or simply benefit) will be defined as the  $\liminf$  of the average benefit because time  $t$  will increase to infinity and the average benefit may not have limiting values.

$$\beta_i(t) = \lim_{t \rightarrow \infty} \inf \beta_i(t) \quad (2.6)$$

### strategies

Because the players move in a stealthy way, there are different types of feedback that a player can get while moving. These types of feedback can be divided into two groups of strategies. The non-adaptive strategies and the adaptive strategies. These are described in table 2.1.

If there is no feedback for either player, we have a non-adaptive strategy. Because a player does not receive any feedback during the game he will play in the same manner against every opponent. The strategy is called non-adaptive because the playing strategy is not dependent on the opponents movements. An interesting subclass of the non-adaptive strategies is the one where the time intervals between two consecutive moves are generated by a renewal process. An example of such renewal strategy is the periodic strategy where the time between two consecutive moves of the players are a fixed interval. An exponential strategy is a renewal strategy in which the interval between two consecutive moves is exponentially distributed. In case there is feedback, a player can adapt his strategy to the information received about the opponent's moves. Depending on the amount of information received, two subclasses of adaptive strategies can be identified. The Last Move (LM) strategies represent the class where whenever a player flips he will find out the exact time that the opponent played the last time. In the second class, called Full History (FH), whenever a player flips he will find out the whole history of the opponent's move.

Categories	Classes of Strategies
Non-adaptive (NA)	Renewal
	- Periodic
	- Exponential
Adaptive (AD)	General non-adaptive
	Last move (LM)
	Full History (FH)

TABLE 2.1: Hierarchy of Classes of strategies in FlipIt

In this paper we restrict ourselves to periodic strategies. This choice is motivated by the fact that in a security game a player (defender or attacker) rarely has information about the moves (last move or full history) of his opponent.

### Results of the FlipIt game

The study of the different strategies by means of FlipIt framework allows to derive a number of interesting results:

- periodic games dominate the other renewal strategies, meaning that it is always advantageous to play periodically against an opponent with a renewal strategy;
- periodic games are disadvantageous against players following a Last Move adaptive strategy;
- if the defender plays with a periodic rate that is fast enough he'll force the attacker to drop out;
- any amount of feedback about the opponent received during the game, benefits to a player.

## 2.4 Extensions on FlipIt

Various possible ways to extend FlipIt have already been proposed. Laszka et al. made a lot of additions and extensions to the original game of FlipIt. For instance Laszka et al. extended the basic FlipIt game to multiple resources. The rationale is that for compromising a system in real life, more than just one resource needs to be taken over. An example is that gaining access to deeper layers of a system may require breaking several passwords. The model is called FlipThem [6]. Laszka et al. also use two ways to flip the multiple resources: the AND and the OR control model. In the AND model the attacker only controls the system if he controls all the resources of the system, whereas in the OR model the attacker only needs to compromise one resource to be in control of the entire system.

Another addition of Laszka et al. to the game of FlipIt [7] is extending the game to also consider non-targeted attacks by non-strategic players. In this game the defender tries to maintain control over the resource that is subjected to both targeted and non-targeted attacks. Non-targeted attacks can include phishing, while targeted attacks may include threats delivered through zero day attack vulnerabilities.

One of the last important additions from Laszka et al. [8] is to consider a game with targeted and non-targeted attacks where the moves made by the attacker do not succeed immediately. This is similar to this paper but it has still some major differences. First the moves by the attacker are still covert but the moves made by the defender are known to the attacker. This means that the attacker knows when the defender plays and can change its strategy depending on the moves of the defender. Our motivation for a defender with stealthy moves is that there is not always an intelligent individual that is behind an APT. Some APT's do not know if the computer is already been recovered. Their purpose is to spread. Not to check if they have already infected. . The second difference is that even though both the targeted and non-targeted attacks do not succeed immediately, the delay is determined differently. For the targeted attack the time till it succeeds is given by an exponential distributed random variable with a known rate. The non-targeted attacks are modelled as a single attacker and the time till it succeeds is given by a Poisson process. In our paper the delay is given by one parameter, that can be the result of any virus propagation model. The third and last difference is that the paper of Laszka has multiple attackers and they try to find the best strategy of the defender against both targeted and non-targeted attacks. The conclusion of this paper is that the optimal strategy for the defender is moving periodically.

beter verwoor-  
den

FlipIt has also been applied to several cases in system security. Researchers explored different applications of FlipIt for real-world problems, like password reset policies, VM refresh, cloud auditing and key rotation [2].

Other authors used the FlipIt game to apply it on a specific scenario. To be able to use the FlipIt game, modifications were required for the FlipIt model. One of the scenarios by Pham [12] was to find out whether a resource was compromised or not by the attacker. This could be verified by the defender, who has an extra move "test" beside the flip move. The basic idea is to test with an extra action if the resource has been compromised or not. This move involves also an extra cost.

A three-player game has also been investigated where the flipit framework of two players is extended by another player. This player represents an insider that trades value information with the attacker [3].

Finally researchers also have investigated the behaviour of humans playing FlipIt. A. Nochenson and Grossklags [11] investigate how people really act when given temporal decisions. They found out that the results improve over time but that they are dependent on gender, age, and other individual difference variables. The result also shows that the participants perform generally better when they have more information about the strategy of the opponent which is a computerized player. Reitter et al. [13] extended the work of A. Nochenson and Grossklags to include va-

rious visual presentation modalities for the available feedback during the investigation.



## Hoofdstuk 3

# FlipIt game with virus propagation

### 3.1 Introduction

The game of FlipIt with virus propagation considers attackers where their moves are not instantaneous. This can be motivated by an example of a virus. A virus can be dropped on a network but it only compromises the whole network if every node in the network is infected. The basic FlipIt game does not take this into account. In this chapter the FlipIt game with virus propagation is explained and how it can be modelled. First the main differences with the basic FlipIt game are discussed in section 3.3. Next in section ?? the formal definition of the game with virus propagation is given. At last, in section 3.4 a periodic strategy game is considered and the formalization of the formulas with virus propagation are determined.

This chapter explains how to model a FlipIt game with a virus propagation that infects a network. The first section explains the difference between a normal FlipIt game and a FlipIt game with virus propagation. The next section derives a formula to calculate the benefit for a FlipIt game with a virus propagation. In the last section we calculate the Nash equilibrium for the benefit formula.

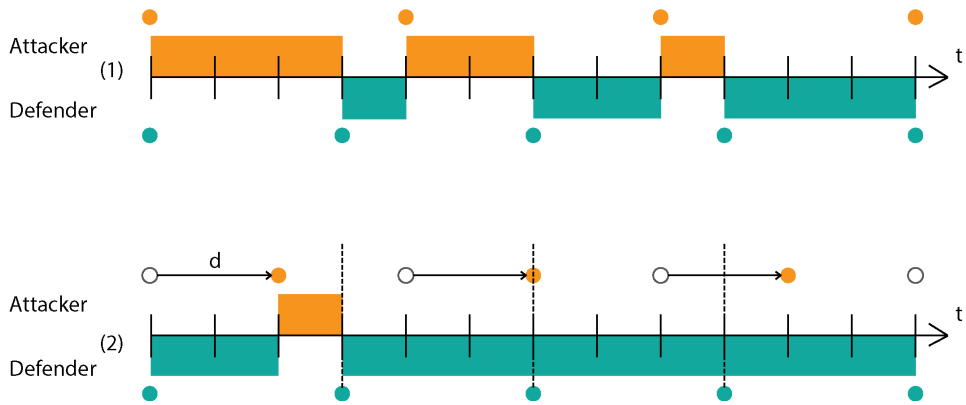
### 3.2 FlipIt game with virus propagation

Welke aanpassing en motivatie: attacker 1 speler die elke keer virus probeert te droppen. Defender die elke keer patcht als er virus gedropt wordt. Elke patch voldoet aan de meest recente patch dus alle virussen die ook gedropt zijn worden weggedaan. Voor future work: patch systeem aanpassen.

The actions of a defender: the actions of an attacker:

### 3.3 Explaining difference between FlipIt with and without virus propagation

FIGUUR 3.1: The first game is the basic FlipIt game. The second is a FlipIt game with a delay. During the first flip of the attacker, the defender moves after the delay, causing the attacker to get control over the resource. During the second flip of the attacker, the defender flips at time  $t+d$ , causing the defender to take control over the resource before the attacker. During the third and final flip of the attacker, the defender flips during time  $t+d$ , causing the attacker to never gain control over the resource.



A FlipIt game consists of a single resource. To represent the security problem, the game now defines its single resource as a computer network with multiple nodes. One of the players, the defender, will try to defend his network. The defender will do this by flipping all the nodes of the network (i.e. the entire resource) in every move he plays. The attacker on the other hand will try to infect all the nodes in the network. The attacker will do this by flipping the node in the graph that can infect all the nodes in the shortest possible time. After dropping a virus on the first node, it takes a while for the virus to infect the entire network. Since the original FlipIt game works with a single resource that is always flipped entirely, the assumption is made that the attacker is considered to gain immediate full control over the resource when the network has been infected, even it is only one node that has been infected.

In reality however, after dropping a virus on the first node, it takes a while for the virus to infect the entire network. So, the assumption that the attacker has full control over the resource as soon as a node has been infected, is not realistic. The attacker has only control of the network once all or a sufficient number of nodes are infected. The time that it takes for the virus to infect every node (or a sufficient number of nodes) will be denoted as an infection-delay variable  $d$  (called 'delay' for short in the remainder of this paper). If we want to measure how long it takes for the virus to infect all the nodes in the network, we have to calculate the shortest path from the first infected node to the farthest node. Rather than denoting the



time needed for infecting *all* the nodes, the variable  $d$  can also be used to denote the time needed to infect *a sufficient number* of nodes.

Assume that an attacker attacks at time  $t$ , he doesn't get immediate control over the resource, but he only gains control at time  $t + d$ , with  $d$  denoting the time needed to infect a sufficiently number (or all) nodes. If the defender flips the network before the period  $d$  has elapsed (so, somewhere between  $t$  and  $t + d$ ), then the attacker will never gain full control over the resource. See figure 3.1. This implies that the mathematical formulas for gain and benefit need to be adapted to the fact that the attacker loses part of its benefit because of this delay. In the remainder of this paper, we will adapt the formalization of the FlipIt game using the variable  $d$ .

### 3.4 Playing periodically with virus propagation

The formalization starts from the model of the non-adaptive continuous basic FlipIt game where players use a periodic strategy with a random phase. This choice is motivated by the assumption that in most organisations, the defence strategy is to periodically defend the network. This corresponds to a periodic defender strategy. A periodic attacker strategy is assumed as well, to be able to compare the results with the periodic strategy of the FlipIt game in [14]. Further research can investigate the effect of relaxing this assumption.

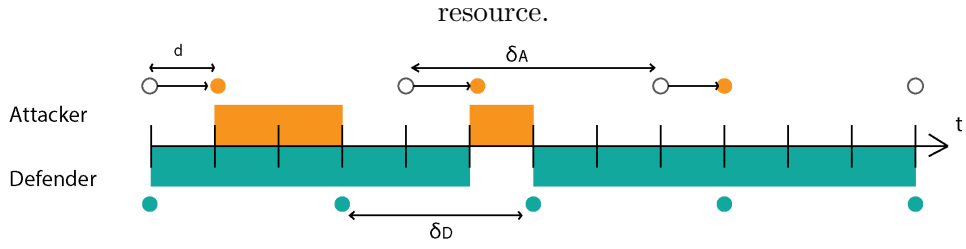
Similarly as in [14], we split the formalization in two cases. The first case is where the defender plays at least as fast as the attacker, the second case is where the attacker plays at least as fast as the defender. For each of these cases, first the benefit formula of the basic case without delay is presented, and then the delay is introduced.

Intuitively we could assume that  $d$  can never be bigger than  $\delta_A$  because then the benefit for the defender would always be 1. This is not always true. It is only true if  $d$  is bigger than  $\delta_D$ , because then the defender will always be in control. For this we only calculate the formulas for the cases where  $d$  is smaller than  $\delta_D$ . We can already conclude that it is no use for the attacker to play when the delay is bigger than  $\delta_D$ .

#### 3.4.1 Formalization the benefit formula including the infection-delay

A Periodic strategy is a non-adaptive renewal strategy where the time intervals between consecutive moves are a fixed period, denoted by  $\delta$ . Moreover it has a random phase, that is chosen uniformly and random in the interval  $[0, \delta]$  for the first move. The average rate of play of a player is denoted by  $\alpha_i = \frac{1}{\delta_i}$ . List of symbols that will be used throughout the paper. Figure 3.2 represents a couple of the symbols for clarification.

FIGUUR 3.2: Formalization of a FlipIt game with delay: A representation of a FlipIt game where both players are playing periodically. Every move or flip is indicated by a blue or orange circle. The defender is represented in blue and plays with a period of  $\delta_D$ . The flip of the attacker is represented by a white circle, but because there is a delay  $d$ , the attacker only controls the resource after time  $d$  represented by an orange circle. The attacker plays with a period of  $\delta_A$ . The blue and orange rectangles represent the amount of time the respective player is in control of the resource.



$i$ : Defines the player. Different as in the FlipIt paper where the defender is denoted by the subscript 0 and the attacker by the subscript 1.

$\delta_i$ : The length of the interval between two consecutive moves of player  $i$ .

$\alpha_i$ : The average flip rate of player  $i$ , given by  $\alpha_i = 1/\delta_i$ .

$k_i$ : The cost of player  $i$ 's moves.

$d$ : The delay caused by the virus propagation.

$G_i(t)$ : The total gain of player  $i$  denotes the amount of time player  $i$  is in control over the resource up to time  $t$ .

$\gamma_i$ : The average gain rate of player  $i$  defined as  $G_i(t)/t$

$\beta_i$ : The average benefit rate up to time  $t$  defined as  $\beta_i = \gamma_i - k_i\alpha_i$ .

$opt_i$ : The optimum function.

**Case 1:  $\delta_D \leq \delta_A$  (The defender plays at least as fast as the attacker.)**

Let  $r = \frac{\delta_D}{\delta_A}$ . The intervals between two consecutive defender's moves have length  $\delta_D$ . Consider a given defender move interval. The probability over the attacker's phase selection that the attacker moves in this interval is  $r$ . Given that the attacker moves within the interval, he moves exactly once within the interval (since  $\delta_D \leq \delta_A$ ) and his move is distributed uniformly at random.

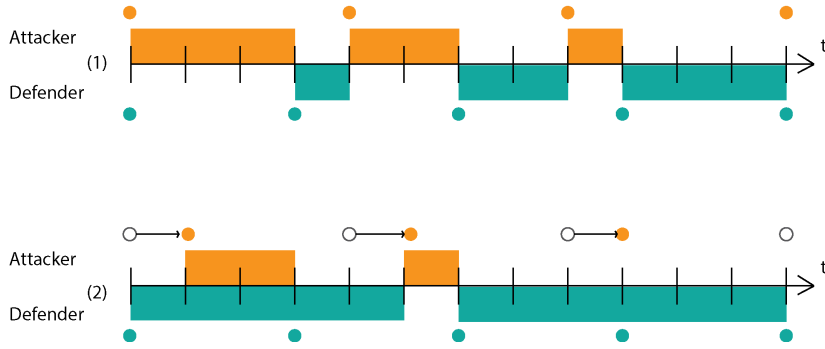
The expected period of attacker control within the interval would be  $r/2$ , without considering the delay by a virus. Therefore the benefit for the attacker, without considering the delay, can be expressed as follows:

$$\beta_A(\alpha_D, \alpha_A) = \frac{r}{2} - k_A \alpha_A = \frac{\delta_D}{2\delta_A} - k_A \alpha_A$$

Correspondingly, the benefit for the defender can be expressed as:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{r}{2} - k_D \alpha_D = 1 - \frac{\delta_D}{2\delta_A} - k_D \alpha_D$$

FIGUUR 3.3: Case 1: Difference between a basic FlipIt game and a FlipIt game with a delay. (1) is the FlipIt game without a delay and (2) is with a delay. The delay is denoted with an arrow. The attacker is only in control when the circle becomes orange.



However, because of the delay required for virus propagation, the maximal time of control is reduced to  $\delta_D - d$ , see figure 3.3. There is a probability of  $r$  that the attacker will move in the interval of the defender. However, the gain will not be half of the interval. Indeed, the attacker has to play soon enough to gain control, meaning that the attacker has to play during the period of  $\delta_D - d$  during the interval of the defender. The probability that the attacker plays soon enough is  $\frac{\delta_D - d}{\delta_D}$  and

this will give the attacker an average gain of  $\frac{\delta_D - d}{2}$ . If the attacker moves after the period of  $\delta_D - d$ , the gain of the attacker will be zero. The probability that this happens is  $\frac{d}{\delta_D}$ . The average gain rate of the attacker can then be expressed as follows if we look at one interval of the defender:

$$\gamma_A(\alpha_D, \alpha_A) = \frac{1}{\delta_D} \left[ \frac{\delta_D}{\delta_A} \cdot \frac{\delta_D - d}{\delta_D} \cdot \frac{\delta_D - d}{2} + \frac{\delta_D}{\delta_A} \cdot \frac{d}{\delta_D} \cdot 0 \right]$$

To derive the benefit, the cost of moving is subtracted from the average gain.

$$\beta_A(\alpha_D, \alpha_A) = \frac{(\delta_D - d)^2}{2 \cdot \delta_D \delta_A} - k_A \alpha_A$$

$$\beta_A(\alpha_D, \alpha_A) = \frac{\delta_D}{2 \cdot \delta_A} - k_A \alpha_A - \left( \frac{d^2}{2 \cdot \delta_A \delta_D} - \frac{d}{\delta_A} \right)$$

The benefit of the defender is expressed as follows:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{(\delta_D - d)^2}{2 \cdot \delta_D \delta_A} - k_D \alpha_D$$

We can easily see that when  $d=0$ , we obtain the formula of the original FlipIt game.

**Case 2:  $\delta_A \leq \delta_D$  (The attacker plays at least as fast as the defender.)**

First let  $r = \frac{\delta_D}{\delta_A}$ . The intervals between two consecutive attacker's moves have length  $\delta_A$ . Consider a given attacker's move interval. The probability over the attacker's phase selection that the defender moves in this interval is  $\frac{\delta_A}{\delta_D} = (1/r)$ . Given that the defender moves within the interval of the attacker, he moves exactly once within this interval (since  $\delta_A \leq \delta_D$ ) and his move is distributed uniformly at random.

A similar analysis as in case 1 for a FlipIt game without virus propagation yields the following benefits:

$$\begin{aligned} \beta_D(\alpha_D, \alpha_A) &= \frac{1}{2r} - k_D \alpha_D = \frac{\delta_A}{2\delta_D} - k_D \alpha_D \\ \beta_A(\alpha_D, \alpha_A) &= 1 - \frac{1}{2r} - k_A \alpha_A = 1 - \frac{\delta_A}{2\delta_D} - k_A \alpha_A \end{aligned}$$

An intuitive solution for the case with a virus would be to subtract the benefit of the attacker received in each interval with the delay similarly as in case 1. This would give us the following formula if we derive it in the same way:

$$\beta_A = \frac{(\delta_A - d)^2}{2\delta_A \delta_D} - \frac{k_A}{\delta_A}$$

But this results in an overestimation. How closer  $\delta_A/\delta_D$  is equal to one, the better the approximation. If  $\delta_A/\delta_D = 1$  the result is correct. This formula does not take into account that if the attacker was in control in the previous interval, the delay should not be subtracted, because the delay will not be in control of the defender. This means that we have to look at what happens in the previous interval.

From the defender we know that his moves are instantaneous. It is easier to calculate the benefit of the defender in this case. Because the defender moves slower than the attacker we know that if the defender moves during the interval of the attacker, he only moves once within this interval. The defender will move during the

interval of the attacker with a probability of  $\frac{\delta_A}{\delta_D}$ . When this happens the defender will end with being in control at the end of the interval. In the next interval the attacker will have to regain control, meaning that during the delay, the defender stays in control, see figure 3.4 cases (1) and (2). The defender will keep the control over the resource in the next interval over a period of the delay, namely  $d$ .

Consider a timespan  $\delta_A + d$ , representing the attacker's interval followed by the delay period in his next interval. The defender will never move twice during this timespan because  $\delta_A + d \leq \delta_D$ . Because  $d + \delta_A \leq \delta_D$  the next move of the defender in this second interval will never occur during the delay, meaning that the entire delay can be considered as an extra benefit resulting of a play in the previous interval. So, every time the defender plays, he will get an average gain of  $\frac{\delta_A}{2}$  in the interval where he plays and in the next interval will always receive a extra gain of  $d$ , yielding

a total average gain per interval of  $\frac{(d + \frac{\delta_A}{2})}{\delta_A}$

For the case with a virus we consider two cases, Case a and Case b, depending on whether the delay is shorter or longer than the difference between the attacker's and the defender's period.

**Case a:**  $d + \delta_A \leq \delta_D$

The total gain rate of the defender is then the probability that the defender will move during an interval of the attacker multiplied by the total average gain per interval:

$$\begin{aligned}\gamma_D(\alpha_D, \alpha_A) &= \frac{\delta_A}{\delta_D} \cdot \frac{(d + \frac{\delta_A}{2})}{\delta_A} \\ \gamma_D(\alpha_D, \alpha_A) &= \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D}\end{aligned}$$

This yields in the following benefit formula:

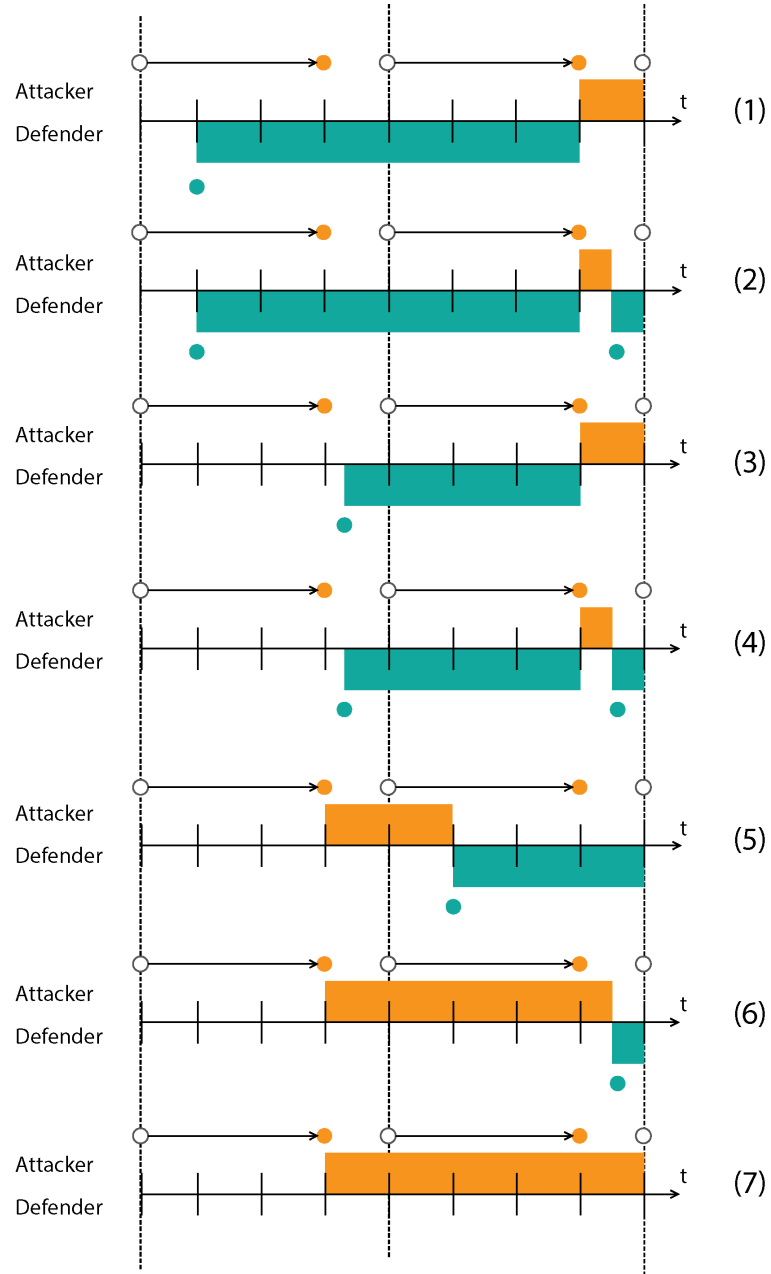
$$\beta_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D \alpha_D$$

The benefit for the attacker will be as follows:

$$\beta_A(\alpha_D, \alpha_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{d}{\delta_D} - k_A \alpha_A$$

It is crucial that  $\delta_D$  is at least as large as  $d + \delta_A$ . If not, this would mean that the defender can move during the delay in the interval following the interval where the

FIGUUR 3.4: All possible cases for the attacker and the defender in Case 2.A where  $d + \delta_A < \delta_D$ . As can be seen, the defender will have control during a period of  $d$  over the resource when the defender has flipped in the previous interval.



defender already moved. This would mean that there can be an overlap between the average gain of  $\frac{\delta_A}{2}$  and the delay. The above benefit formula would then include to much gain for the defender: the potential overlap during the delay would be counted twice.

**Case b:**  $d + \delta_A \geq \delta_D$

To obtain the formula in case of a too long delay, we therefore need to subtract this overlapping gain from the above formula. Since  $\delta_D \geq \delta_A$ , if the defender enters the interval immediately after the attacker has played, then the defender cannot have played in the previous interval. In that case, there is no overlap. So the problem of the overlap only appears if the defenders enters late enough and thus only the last part of the delay is subject to overlap. The larger the difference between the interval of the defender and the attacker, the smaller the risk of overlap. Concretely, only the last part of length  $d - (\delta_D - \delta_A)$  is subject to overlap. Hence, the probability of overlap is  $\frac{d - (\delta_D - \delta_A)}{\delta_D}$  and the gain will be half of this interval:  $\frac{d - (\delta_D - \delta_A)}{2}$ . The gain rate to be subtracted is therefore:

$$\frac{1}{\delta_A} \cdot \frac{d - (\delta_D - \delta_A)}{\delta_D} \cdot \frac{d - (\delta_D - \delta_A)}{\delta_D}$$

The total gain rate of the defender is obtained by subtracting this term from the gain rate of case a:

$$\begin{aligned} \gamma_D(\alpha_D, \alpha_A) &= \frac{\delta_A}{\delta_D} \cdot \frac{(d + \frac{\delta_A}{2})}{\delta_A} - \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A} \\ \gamma_D(\alpha_D, \alpha_A) &= \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A} \end{aligned}$$

This yields in the following benefit formula:

$$\beta_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D\alpha_D - \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A}$$

The benefit for the attacker will be as follows:

$$\beta_A(\alpha_D, \alpha_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{d}{\delta_D} - k_A\alpha_A + \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A}$$





## Hoofdstuk 4

# Nash Equilibria

### 4.1 Nash Equilibria

– rechtstreeks uit FlipIt paper –

As a second step, we are interested in finding Nash equilibria, points for which neither player will increase his benefit by changing his rate of play. More formally, a Nash equilibrium for the periodic game is a point  $(\alpha_0^*, \alpha_1^*)$  such that the defender's benefit  $\beta_0(\alpha_0, \alpha_1^*)$  is maximized at  $\alpha_0 = \alpha_0^*$  and the attacker's benefit  $\beta_1(\alpha_0^*, \alpha_1)$  is maximized at  $\alpha_1 = \alpha_1^*$ . To begin with, some useful notation. We denote by  $\text{opt0}(\alpha_1)$  the set of values (rates of play  $\alpha_0$ ) that optimize the benefit of the defender for a fixed rate of play  $\alpha_1$  of the attacker. Similarly, we denote by  $\text{opt1}(\alpha_0)$  the set of values (rates of play  $\alpha_1$ ) that optimize the benefit of the attacker for a fixed rate of play  $\alpha_0$  of the defender. The following theorem specifies Nash equilibria for the periodic game and is proven in Appendix A.

#### 4.1.1 Determining the piecewise functions $\text{opt}_D(\delta_A)$

Nash equilibria are points with the property that neither player benefits by deviating in isolation from equilibrium. We can compute Nash Equilibria for the periodic game as an intersection points of curves  $\text{opt0}$  and  $\text{opt1}$ . To determine  $\text{opt0}(\alpha_1)$  we need to compute the derivative of  $\beta_0(\alpha_0, \alpha_1)$  for a fixed  $\alpha_1$ . We consider two cases:

##### Case 1: $\delta_D \leq \delta_A$

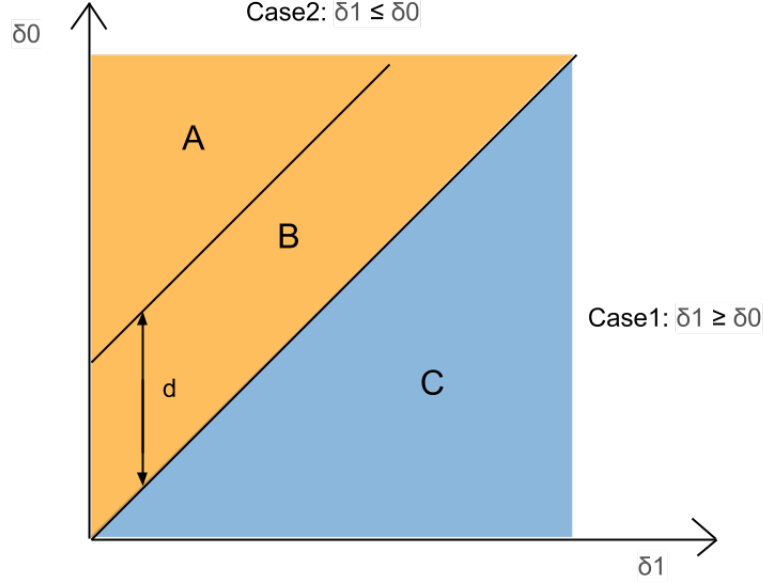
The benefit formula obtained previously for this case is as follows:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{\delta_D}{2\delta_A} - \frac{k_D}{\delta_D} - \frac{d^2}{2\delta_D\delta_A} + \frac{d}{\delta_A}$$

If we take the partial derivative for a fixed  $\delta_1$  we get the following result:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = -\frac{1}{2\delta_A} + \frac{k_D}{\delta_D^2} + \frac{d^2}{2\delta_D^2\delta_A}$$

FIGUUR 4.1: This figure shows the three subcategories of each case. 'A' stands for Case 2.A:  $\delta_D \geq d + \delta_A \geq \delta_A$  and 'B' stands for Case 2.B:  $d + \delta_A \geq \delta_D \geq \delta_A$



The stationary points (maximum, minimum) can be found by setting the first derivative equal to zero and finding the roots of the resulting equation:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad \delta_D = \sqrt{2\delta_A k_D + d^2}$$

This leads to the following deduction: The function increases on  $[0, \sqrt{2\delta_A k_D + d^2}]$  and is decreasing on  $[\sqrt{2\delta_A k_D + d^2}, \infty]$ . So we got a maximum on  $\delta_D = \min\{\delta_A, \sqrt{2\delta_A k_D + d^2}\}$ . The minimum of the two values is needed because  $\delta_D$  cannot be bigger than  $\delta_A$ .

**Case 2.A:**  $\delta_D \geq d + \delta_A \geq \delta_A$

The benefit formula obtained previously for this case is as follows:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - \frac{k_D}{\delta_D}$$

The derivative of the above formula for a fixed  $\delta_A$  results in the following:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = -\frac{\delta_A}{2\delta_D^2} - \frac{d}{\delta_D^2} + \frac{k_D}{\delta_D^2}$$

To obtain the stationary points the first derivative is set equal to zero and the roots of the resulting equation are found:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad \delta_A = 2(k_D - d) = dk_D - 2d$$

This leads to the following deduction:

If  $k_D \leq d$

decreasing  $\delta_A < 2(k_D - d)$

increasing  $\delta_A > 2(k_D - d)$

If  $k_D > d$

increasing  $\delta_A < 2(k_D - d)$

decreasing  $\delta_A > 2(k_D - d)$

**Case 2.B:**  $d + \delta_A \geq \delta_D \geq \delta_A$

The benefit formula obtained previously for this case is as follows:

$$\frac{\beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - \frac{k_D}{\delta_D} - \frac{(d - (\delta_D - \delta_A))^2}{d\delta_D\delta_A}$$

The derivative of the above formula for a fixed  $\delta_A$  results in the following:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = -\frac{1}{2\delta_D} + \frac{k_D}{\delta_D^2} + \frac{d^2}{2\delta_D^2\delta_A}$$

The stationary points (maximum, minimum) can be found by setting the first derivative equal to zero and finding the roots of the resulting equation:

$$\frac{\partial \beta_D(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad \delta_D = \sqrt{2\delta_A k_D + d^2}$$

This leads to the following deduction: The function increases on  $[0, \sqrt{2\delta_A k_D + d^2}]$  and is decreasing on  $[\sqrt{2\delta_A k_D + d^2}, \infty]$  So we got a maximum on  $\delta_D = \text{minimum}\{\delta_A, \sqrt{2\delta_A k_D + d^2}\}$

### Best responses

The optimum functions will be piecewise functions. There will be different optimum functions depending on  $k_D$  and  $d$ . We distinguish for these two cases, three sub cases for different values of  $\delta_A$ .

$$k_D \leq d$$

Because  $k_D \leq d$ , the term  $2(k_D - d)$  will always be negative. We point out that  $\delta_A$  and  $\delta_D$  are positive rates.

- if  $\delta_A < 2(k_D - d)$   
This means that  $\delta_A$  has to be negative which is not possible. For case the defender will not play.
- $\delta_A = 2(k_D - d)$   
 $\delta_A$  is negative or equal to 0 so the attacker will not play. For case 1 and case 2.b the defender will also not play.
- $\delta_A > 2(k_D - d)$   
Case 2.a it is increasing for every value  $\delta_A \in [0, \infty]$ . For case 1 together with case 2.b the optimal benefit is achieved at rate  $\delta_D = \sqrt{d^2 + 2\delta_A k_D}$ .

$$k_D > d$$

Because  $k_D > d$ , the term  $2(k_D - d)$  will always be positive. We point out that  $\delta_A$  and  $\delta_D$  are positive rates.

- if  $\delta_A < 2(k_D - d)$   
From case 2.a it follows that the benefit of the defender increases. From case 1 and case 2.b together the optimal benefit of the defender is achieved at rate  $\delta_D = \sqrt{d^2 + 2\delta_A k_D}$ .
- $\delta_A = 2(k_D - d)$   
From case 2.a it follows that  $\beta_D(\delta_D, \delta_A) = 0$ , for all  $\delta_A \in [0, 2(k_D - d)]$ . From case 1 and case 2.b together the optimal benefit for the defender is achieved for all rates  $\delta_D \in [0, \sqrt{d^2 + 2\delta_A k_D}]$ .
- $\delta_A > 2(k_D - d)$   
For case 2.a the benefit is decreasing. From case 1 and case 2.b the best strategy for the defender is not playing at all.

dat laatste nog eens nakijken

From this analyses we can compute  $opt_D(\delta_A)$  for two different cases as:

$$opt_D(\delta_A) = \begin{cases} 0, & \delta_A < 2(k_D - d) \\ 0, & \delta_A = 2(k_D - d) \\ \sqrt{d^2 + 2\delta_A k_D}, & \delta_A > 2(k_D - d) \end{cases}$$

For case :

$$opt_D(\delta_A) = \begin{cases} \sqrt{d^2 + 2\delta_A k_D}, & \delta_A < 2(k_D - d) \\ [0, \sqrt{d^2 + 2\delta_A k_D}], & \delta_A = 2(k_D - d) \\ 0, & \delta_A > 2(k_D - d) \end{cases}$$

### 4.1.2 Determining the piecewise functions $opt_A(\delta_D)$

We still consider the case where  $d < \delta_D$ .

To determine the Nash equilibria we also need to determine  $opt_A(\delta_D)$  by computing the derivative of  $\beta_A(\delta_D, \delta_A)$  for a fixed  $\delta_D$ . We consider 2 cases:

#### Case 1: $\delta_A \geq \delta_D$

The benefit formula obtained previously for this case is as follows:

$$\beta_A(\delta_D, \delta_A) = \frac{\delta_D}{2\delta_A} - \frac{k_A}{\delta_A} + \frac{d^2}{2\delta_D\delta_A^2} - \frac{d}{\delta_A}$$

The derivative for a fixed  $\delta_D$  is as follows:

$$\frac{\partial \beta_A(\delta_D, \delta_A)}{\partial \delta_A} = -\frac{\delta_D}{2\delta_A^2} + \frac{k_A}{\delta_A^2} - \frac{d^2}{2\delta_D\delta_A^2} + \frac{d}{\delta_A^2}$$

The stationary points (maximum, minimum) can be found by setting the first derivative equal to zero and finding the roots of the resulting equation:

$$\frac{\partial \beta_A(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad 2k_A = \frac{(\delta_D - d)^2}{\delta_D}$$

It follows that  $\beta_A(\delta_D, \cdot)$  is increasing if  $2k_A < (\delta_D - d)^2/\delta_D$  and decreasing if  $2k_A > (\delta_D - d)^2/\delta_D$ .

#### Case 2.A: $\delta_D \geq d + \delta_A \geq \delta_A$

The benefit formula obtained previously for this case is as follows:

$$\beta_A(\delta_D, \delta_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{k_A}{\delta_A} - \frac{d}{\delta_D}$$

The derivative for a fixed  $\delta_D$  is as follows:

$$\frac{\partial \beta_A(\delta_D, \delta_A)}{\partial \delta_A} = \frac{-1}{2\delta_D} + \frac{k_A}{\delta_A^2}$$

The stationary points (maximum, minimum) can be found by setting the first derivative equal to zero and finding the roots of the resulting equation:

$$\frac{\partial \beta_A(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad \delta_A = \sqrt{2\delta_D k_A}$$

it follows that  $\beta_A(\delta_D, \cdot)$  is increasing on  $[0, \sqrt{2k_A\delta_D}]$  and decreasing on  $[\sqrt{2k_A\delta_D}, \infty]$  and thus has a maximum on  $\delta_A = \max\{\delta_D, \sqrt{2k_A\delta_D}\}$ . The maximum between  $\delta_D$  and  $\sqrt{2k_A\delta_D}$  is needed because  $\delta_A$  cannot exceed  $\delta_D$  in this case.

**Case 2.B:**  $d + \delta_A \geq \delta_D \geq \delta_A$

The benefit formula obtained previously for this case is as follows:

$$\beta_A(\delta_D, \delta_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{d}{\delta_A} - \frac{k_A}{\delta_A} + \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A}$$

The derivative for a fixed  $\delta_D$  is as follows:

$$\frac{\partial \beta_A(\delta_D, \delta_A)}{\partial \delta_A} = -\frac{\delta_D}{2\delta_A^2} + \frac{k_A}{\delta_A^2} - \frac{d^2}{2\delta_D\delta_A^2} + \frac{d}{\delta_A^2}$$

The stationary points (maximum, minimum) can be found by setting the first derivative equal to zero and finding the roots of the resulting equation:

$$\frac{\partial \beta_A(\alpha_D, \alpha_A)}{\partial \alpha_D} = 0 \quad \Rightarrow \quad 2k_A = \frac{(\delta_D - d)^2}{\delta_D}$$

it follows that  $\beta_A(\delta_D, \cdot)$  is increasing if  $2k_A < (\delta_D - d)^2/\delta_D$  and decreasing if  $2k_A > (\delta_D - d)^2/\delta_D$ . This is the same result as in case 1.

**Best responses**

The optimum functions will be piecewise functions. We distinguish three cases for different values of  $\delta_d$  and  $k_A$ .

For this term  $\frac{(\delta_D - d)^2}{\delta_D}$ ,  $d$  has to be bigger than  $\delta_D$  because the cost  $k_A$  cannot be negative. This was an assumption that was already made, because the benefit of the defender will always be 1 if  $d$  is bigger than  $\delta_D$ .

- if  $2k_A < \frac{(\delta_D - d)^2}{\delta_D}$   
Then for case 1 and case 2.b the benefit of the defender is increasing. From case 2.a follows that the optimal benefit for the attacker is achieved at the rate  $\delta_A = \delta_D$
- if  $2k_A = \frac{(\delta_D - d)^2}{\delta_D}$   
From case 1 and case 2.b it follows that  $\beta_D(\delta_D, \delta_A) = 0$ , for all  $\delta_A \in [0, \frac{(\delta_D - d)^2}{2k_A}]$ . From case 2.a the optimal benefit for the defender is achieved for all rates  $\delta_A \in [0, \delta_D]$ .
- if  $2k_A > \frac{(\delta_D - d)^2}{\delta_D}$   
All decreasing.

This brings us to the following opt function for : From this analyses we can compute  $opt_D(\delta_A)$  for two different cases as:

$$opt_D(\delta_A) = \begin{cases} 0, & \delta_A < 2(k_D - d) \\ 0, & \delta_A = 2(k_D - d) \\ \sqrt{d^2 + 2\delta_A k_D}, & \delta_A > 2(k_D - d) \end{cases}$$





## Hoofdstuk 5

# Virus Propagation

### 5.1 Methods of virus propagation

The spreading of worms has already been extensively researched [beter schrijven]. Because of all the different types of propagation it is hard to pick out one model that can model all of them. Modelling the spread of malware depends on two great factors: the method used for the propagation and the graph where the malware will spread itself. Viruses and worms are the types of malware that are the most researched. Under we list a couple of examples to model virus and worm propagation. In this text worm and virus will be used intermediate.

#### 5.1.1 Models of graph representing the network

##### Methods of propagation

Random scanning: Some worms propagate by using the method of random scanning IP addresses. Rate of terms of succeeded randwomly choosen IP addresses -> maximise good ip addresses. After a while the remaining nodes will be les likely to be reached. An example of such a worm is 'Code Red'

Kinds of worms: random scanning, email worm internet worms.

Routing worms: use BGP routing tables to scan the routable address space. 3 times faster than traditional worm. Spyb0t or network bluepill. Another propagation method: worms that stop for a while and then continue. Atak worm Stackelberg game: first move is from attacker, defender is follower.

Kind of network model: Kind of model: SIS, SIR

Popular mechanism that worms use to detect vulnerable targets by random ip scanning probing. Feasible due to use 32-bit addresses. 128-bit addresses life harder for worms, except the ones that use email systems to propagate. two new strategies: uniformly distributed random number generator to select new target. :spread locally, by biasing the search space towards addresses within the same subnet or network. The second strategy is almost the same as what email virusses would get. For this reason we work an example out ..

A method to calculate the propagation of the virus in an easy way. Google page ranking algorithm.

### 5.2 REpresenting Virus Propagation in a network

### 5.3 Related work

Verschillende soorten threats: known, unknown, targeted. APT is het soort threat waar er zeker naar gekeken moet worden. Ge weet niet wanneer ze toeslaan. Normaal gezien bestaat de security van een bedrijf in verschillende lagen. Malware detection, firewalls, application whitelisting, .. Maar APT glippen daar tussendoor. Het kost het bedrijf veel minder om zich ertegen te verdedigen dan alles op te lossen als een aPT heeft kunnen toeslaan. Voorbeelden van APT's die uiteindelijk ontdekt zijn en welke schade ze hebben toegegaan.

– Related work –

There are various different ways for a virus to propagate through a network. Virus, what is a virus ?

The "I love you"virus is an example of a virus that spreads quickly. This virus propagates via mail systems. If someone opens an email with "I love you"virus in annex this virus spreads itself by sending a mail itself to everyone in your contact list. So the virus can multiply rapidly and eventually a business network shut down by the heavy traffic. In this example, there is a need human interaction to spread the virus to do. If no one opens the virus can not spread the mail. Unfortunately, there are viruses that can spread without human interaction. These viruses are referred to as worms. A worm is also a computer program that replicates itself to spread to other computers so. Via a computer network, copies of the worm forwarded without an intermediary is used for. The worm will use vulnerabilities to infect other computers. Most worms are designed to spread out and just try not to make any changes to the systems that they pass. These worms can still inflict damage by increased network traffic they generate. Worms that contain Harm damage a program to install a backdoor or a rootkit on the infected computers. Backdoors and rootkits ensure that future use can be made of the infected computers. The Stuxnetworm is a very famous worm. Initially this worm spread via infected USB sticks and from then it could spread through the Internet to other computers. The purpose of the Stuxnetworm was broken to run the centrifuges in nuclear reactors. Many reactors have been infected. From the standpoint of the defender, it is very important to respond as quickly as possible so that the worm can not spread quickly.

### 5.4 Representing Virus Propagation in a network

### 5.5 methode met matrixen

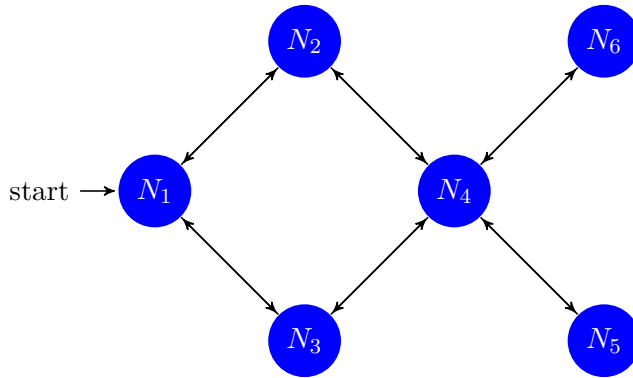
source: [http://en.wikipedia.org/wiki/Adjacency\\_matrix](http://en.wikipedia.org/wiki/Adjacency_matrix)

We model the network through an undirected Graph  $G = \langle V, E \rangle$  where  $|V|$  denotes

the number of resources in the network and  $|E|$  the number of connections. We can convert this to a adjacent matrix where we can represent which vertices of the graph are neighbours of other vertices.

For our graph we have an  $|V| \times |V|$  matrix with on every entry  $a_{ij}$  a 1 as value if there is a connection between node  $V_i$  and  $V_j$  and with zeros its diagonal. Because our graph is undirected we have a symmetric matrix.

*"If  $A$  is the adjacency matrix of the directed or undirected graph  $G$ , then the matrix  $A^n$  (i.e., the matrix product of  $n$  copies of  $A$ ) has an interesting interpretation: the entry in row  $i$  and column  $j$  gives the number of (directed or undirected) walks of length  $n$  from vertex  $i$  to vertex  $j$ . If  $n$  is the smallest nonnegative integer, such that for all  $i, j$ , the  $(i, j)$ -entry of  $A^n > 0$ , then  $n$  is the distance between vertex  $i$  and vertex  $j$ ." [Wikipedia]*



The adjacent matrix becomes this matrix  $[A]$ :

$$\begin{matrix} & N_1 & N_2 & N_3 & N_4 & N_5 & N_6 \\ \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Matrix  $A \times A = A^2$  becomes the matrix with the number of paths with 2 steps from  $N_i$  to  $N_j$ : We denote this matrix as matrix  $[B]$

$$\begin{matrix} & N_1 & N_2 & N_3 & N_4 & N_5 & N_6 \\ \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{matrix} & \begin{pmatrix} 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & 1 & 1 \\ 0 & 2 & 2 & 0 & 1 & 1 \\ 2 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Matrix  $A^2 \times A = A^3$  becomes the matrix with the number of paths with 3 steps from  $N_i$  to  $N_j$ : We denote this matrix as matrix  $[C]$

$$\begin{array}{c}
N_1 \quad N_2 \quad N_3 \quad N_4 \quad N_5 \quad N_6 \\
\begin{array}{c} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{array} \begin{pmatrix} 0 & 4 & 4 & 0 & 2 & 2 \\ 4 & 0 & 0 & 6 & 0 & 0 \\ 4 & 0 & 0 & 6 & 0 & 0 \\ 0 & 6 & 6 & 0 & 4 & 4 \\ 2 & 0 & 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 4 & 0 & 0 \end{pmatrix}
\end{array}$$

So for  $A^N$  every  $a_{ij}$  entry gives the number of paths with N steps from  $N_i$  to  $N_j$ .

With this knowledge we can calculate in how many steps a node is infected.  $A$  calculates which nodes are infected after 1 step,  $A^N$  calculates which nodes are infected in N steps.. So if we want to know how many nodes are infected after 3 steps we have to add every matrix ( $A + A^2 + A^3$ ) and see which entry is a non zero entry.

What do we need for an algorithm

Graph network  $G = \langle V, E \rangle$

Graph matrix  $[A]$  which is  $|V| \times |V|$

Attack vector  $[X]$  which is  $1 \times |V|$

cummulative matrix  $[M]$  which is  $|V| \times |V|$

state matrix  $[T]$  which is  $|V| \times |V|$

Reset vector  $[R]$

duration  $d$

time  $n$

rate  $\delta_0$  of defender and  $\delta_1$  of attacker

Initialisation algorithm:

```

initialisatie
d=0
A=basismatrix
M=A^{0}
n=0
\delta_{0}
\delta_{1}
X
R
controller = defender

```

Algorithm

$n := n + 1;$

Check who is in control? ( through modulo )

if ( defender & controller=defender)

$d := d + 1;$

if ( defender & controller=attacker )

$G = X \times R$  (flippen ten voordele van defender)

$d = 0$

controller = defender

if ( attacker & controller=defender )

controller=attacker

..

if ( attacker & controller=attacker )

$d := d + 1$

$M = M \times A$

$T = T + M$

$G = X \times T$



# Hoofdstuk 6

## Related work

### 6.1 Related work

Some related work, al dan niet game theorie

Distributed Worm Simulation with a Realistic Internet 2005

Modelling of congestions of network through worm propagation. Mathematical model focussing on the underlying network infrastructure.(diff no game theory)

Of threats and Costs: A Game-theoretic approach to security risk management 2013

Model network security of networks with a non-cooperative node through game theory. Attacker knows the defence strategies and the defender has knowledge of the possible attacks. Each actor considers the actions of the other before deciding to strive to optimize their own utility. (diff not stealthy)

Game theory meet network security and privacy (2013)

Chapter 3 addresses several games in game theory for modelling network security.

Game theoretic approach for cost-benefit analysis of malware proliferation prevention (..) Introduces SIS and SIR together with 'patch', 'removal' and 'patch and removal'.

### 6.2 Why Game Theory to model security problem

Actors in a security protocol must follow the systems and some arbitrarily actors that are malicious and do not follow the protocol. [Bridging Game Theory and Cryptography]. Game theoretic approach proposes a model where all the actors act with self-interest.





## Hoofdstuk 7

# Conclusion

The final chapter contains the overall conclusion. It also contains suggestions for future work and industrial applications.

### 7.1 trala



# Bibliografie

- [1] Standaard: Snowden bevestigt dat belgacom gehackt werd, 17/05/15.
- [2] K. Bowers, M. van Dijk, R. Griffin, A. Juels, A. Oprea, R. Rivest, and N. Triandopoulos. Defending against the unknown enemy: Applying flipit to system security. In J. Grossklags and J. Walrand, editors, *Decision and Game Theory for Security*, volume 7638 of *Lecture Notes in Computer Science*, pages 248–263. Springer Berlin Heidelberg, 2012.
- [3] X. Feng, Z. Zheng, P. Hu, D. Cansever, and P. Mohapatra. Stealthy attacks meets insider threats: A three-player game model.
- [4] Kaspersky. A business approach to managing data security threats. In *IT security risk survey 2014*, 2014.
- [5] Kaspersky. Special report on mitigation strategies for advanced threats. In *The power of protection*, 2014.
- [6] A. Laszka. Flipthem: Modeling targeted attacks with flipit for multiple resources. *5th International Conference, GameSec 2014, Los Angeles, CA, USA, November 6-7, 2014. Proceedings*, 8840:175–194, 2014.
- [7] A. Laszka, B. Johnson, and J. Grossklags. Mitigating covert compromises. *iets*, 8289:319–332, 2013.
- [8] A. Laszka, B. Johnson, and J. Grossklags. Mitigation of targeted and non-targeted covert attacks as a timing game. 8252:175–191, 2013.
- [9] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers, 2008.
- [10] Y. S. M. Jackson, K. Brown. Coursera game theory. Stanford University and The university of British Columbia, 2004.
- [11] A. Nochenson, J. Grossklags, et al. A behavioral investigation of the flipit game. In *Proceedings of the 12th Workshop on the Economics of Information Security (WEIS)*, 2013.

- [12] V. Pham and C. Cid. Are we compromised? modelling security assessment games. In J. Grossklags and J. Walrand, editors, *Decision and Game Theory for Security*, volume 7638 of *Lecture Notes in Computer Science*, pages 234–247. Springer Berlin Heidelberg, 2012.
- [13] D. Reitter, J. Grossklags, and A. Nochenson. Risk-seeking in a continuous game of timing. In *Proceedings of the 13th International Conference on Cognitive Modeling (ICCM)*, pages 397–403, 2013.
- [14] M. van Dijk, A. Juels, A. Oprea, and R. Rivest. Flipit: The game of ”stealthy takeover”. *Journal of Cryptology*, 26(4):655–713, 2013.

## Fiche masterproef

*Student:* Sophie Marien

*Titel:* Flip the virus: Modelling targeted attacks using FlipIt with propagation delay

*Engelse titel:* Flip the virus: Modelling targeted attacks using FlipIt with propagation delay

*UDC:* 621.3

*Korte inhoud:*

Recently, high profile targeted attacks such as the attack on Belgacom (a major Belgian telcom), have demonstrated that even the most secure companies can still be compromised, and that moreover such attacks can go undetected for a while. These kind of attacks are called APT, advanced persistent threats, which are designed to penetrate secretly a computer network, collect sensitive data and stay hidden for many years. A company has every interest to mitigate the risks of an APT and the consequences that it can cause. Because of the stealthiness fighting against these attacks requires methods that go beyond the standard tools against malware. A group of researchers at the RSA, van Dijk et al., proposed the game FlipIt (The game of “stealthy takeover”) to model stealthy takeovers. It is a 2-players game composed of a single attacker, a single defender and a single shared resource. The players will compete to get control over the shared resource. Every move of the players will involve a cost and these moves happen in a stealthy way. The objective of the game for each player is to maximise the fraction of time of controlling the resource and minimise the total move cost.

FlipIt does however not take into account that a move may not be instantaneous, but has a certain delay. We adapt FlipIt such that we can use it to model the game of defending a company network that is attacked by a virus. The FlipIt formulas are adapted such as to take the delay for virus propagation into account, which in our case will be a delay for the attacker. In this paper, we restrict ourselves to games where both the defender and the attacker play with a periodic strategy. The goal of this thesis is to find out if there are interesting Nash Equilibria for a game with a virus propagation delay and if we can learn some lessons out of it.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdspecialisatie Veilige software

*Promotor:* Prof. dr. T. Holvoet

*Assessoren:* Prof. dr. B. Jacobs

Dr. ir. A. Dries

*Begeleiders:* Ir. Jonathan Merlevede,

Ir. Kristof Coninx