KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Flip the virus: Modelling targeted attacks using FlipIt 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. ir. Tom Holvoet

Assessoren:
Ir. W. Eetveel
W. Eetrest

Begeleider:
Ir. Jonathan Merlevede, Ir. Kristof Coninx

Academiejaar 2014 – 2015

Voorwoord

I would like to thank everybody who kept me busy the last year, especially my promotor and my assistants.

Sophie Marien

Inhoudsopgave

Todo list

Samenvatting

There are many possible ways to attack a company network. Everyday they suffer frrom multiple attakcs and stealthy attacks. We will make use of a gamemodel FlipIt to find out what the best strategies are for a network manager to defend his network. A worm or a virus will propagate through the network and will cause nodes to be infected. By flipping it the network manager can keep his network clean. In this thesis I present a work of gametheory merged with cybersecurity. The `abstract` environment contains a more extensive overview of the work. But it should be limited to one page.

Samenvatting

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

Lijst van tabellen

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 possible be abused by competitors through data leakage. Also, disruption caused by distributed DoS attacks, may results 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 frequently occurring threats are Advanced Persistent Threats (APT). An APT is a targeted cyber attack that targets organisations in a stealthy way and that can stay undetected for a long period. This makes it so hard to protect a network or a system against an APT. Bruce Schneier describes an APT as something different and stronger than a conventional threat: "*A conventional hacker or criminal isn't interested in any particular target. He wants a thousand credit card numbers for fraud, or to break into an account and turn it into a zombie, or whatever. Security against this sort of attacker is relative; as long as you're more secure than almost everyone else, the attackers will go after other people, not you. An APT is different; it's an attacker who - for whatever reason - wants to attack you. Against this sort of attacker, the absolute level of your security is what's important. It doesn't matter how secure you are compared to your peers; all that matters is whether you're secure enough to keep him out*" - Bruce Schneier [1].

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. Game theory analyses the security problem as a game where the players are an attacker and a defender of a system, and where both players have to make decisions. In particular, both players will aim for the strategy that results in a maximal benefit for them. Researchers at RSA, Van Dijck et al. , made a game theoretic framework to model targeted attacks. They study the specific scenario where a system or network is repeatedly taken over completely by an attacker and this attack is not immediately detected by the defender of the system or network. In game theory, such a game is known as "FlipIt" [10]. This is a two players game where the attacker and the defender are competing to get control over a shared resource. Both players do not know who is currently in control of the resource until they move. In FlipIt every move gives them immediately 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

The game theoretical approach of the FlipIt does not take such delay into account. This an lead us to the following research questions:

- Is it possible to incorporate the notion of delay in the game-theoretical analysis of the Flip-It game ?

- Does this allow us to determine an optimal defense strategy against an attacker ? for example: Gaat er een specieke grootte zijn van een delay waarbij de attacker al weet dat hem niet meer moet gaan spelen ? (is niet gelijk aan de grootte van de periode van de attacker)

Resarch questions when working with a network and a delay: .. graph model en uitleggen hoe we de graph kunnen maken zodat de delay altijd zo groot mogelijk gaat zijn.

- How can we calculate the expected duration for a node's infection/the entire network infection ?

- Can we calculate this node per node ?

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

–opbouw van de thesis uitleggen–

The organisation of this paper is the following. In chapter 2 a brief introduction to Gametheory is introduced to get familiar with 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. This chapter ends with simulations where conclusions can be derived. Next in Chapter 4 we given 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 5 , we discuss the main results and complications and provide directions for further research.

Hoofdstuk 2

Intoduction to GameTheory

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 [6] and [2] is given in section 2.1 . For a more detailed and full introduction to game theory, the reader is referred to [6]. In section 2.2 an overview of the FlipIt game is given with the definitions and concepts that will be used throughout the paper. The last section 2.3 will cover the extensions and additions already made on FlipIt.

2.1 A brief introduction in Game Theory

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 as cooperative game theory.

List of terms

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

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 he can do.

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

Utility function: The utility function 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.

strategien en acties definieren

2.1.1 Best response and Nash Equilibrium

One of the solution concepts in Game Theory for non-cooperative games is a Nash Equilibrium that we will use in this paper. A Nash Equilibrium is a subset of outcomes that can be interesting to analyse a game. 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. In general we can say that a Nash Equilibrium is a stable strategy profile: each player is considered to know the equilibrium strategies of the other players and no player would want to change his own strategy if he knows the strategies of the other players.

Formal definition of a Nash Equilibrium: A strategy profile $s = (s1, ..., sn)$ is a Nash equilibrium if, for all agents i, si is a best response to $s - i$. "Intuitively, a Nash equilibrium is a stable strategy profile: no agent would want to change his strategy if he knew what strategies the other agents were following. We can divide Nash equilibria into two categories, strict and weak, depending on whether or not every agent's strategy constitutes a unique best response to the other agents' strategies."

Nash beter uitleggen nog met best response erbij

POSTCONDITION: Uitgelegd: Strategien, acties, strategien, spelers, rationeel, Nash, best response

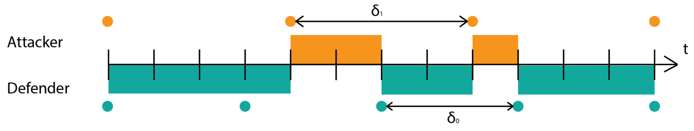
2.2 The FlipIt game

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 ($t \to \infty$). The time in the game is assumed as being continuous, but a discrete time could also be considered. To get control over the resource, the players i, with $i \in \{A, D\}$, can flip the resource at any given time. A flip will be regarded as a move from a player i. Each move will imply a certain cost k_i and the cost can vary for each player. Both players will try to minimize their cost. Adding a cost will prevent players to move too frequently.

The unique feature of FlipIt is that every move will happen 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 will not find out if the resource has already 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.



FIGUUR 2.1: A representation of a FlipIt game where both players are playing periodically and at discrete time intervals. 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.

The state of the resource is denoted 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, $C_A(t)$ will be 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:

$$G_i(t) = \int_0^t C_i(x)dx. \tag{2.1}$$

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

$$G_D(t) + G_A(t) = t \tag{2.2}$$

The average gain rate of player i is defined as:

$$\gamma_i(t) = G_i(t)/t. \tag{2.3}$$

And thus for all $t > 0$:

$$\gamma_D(t) + \gamma_A(t) = 1 \tag{2.4}$$

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

$$\beta_i(t) = \gamma_i(t) - k_i\alpha_i. \tag{2.5}$$

This is equal to the fraction of time the resource has been owned by player i, minus the cost of making the moves. α_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 lim inf of the average benefit because time t will increase to infinity and the average benefit may not have limiting values.

$$\beta_i(t) = \lim_{t\to\infty} inf\beta_i(t) \tag{2.6}$$

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 neither of the players, 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

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

Tabel 2.1: Hierarchy of Classes of strategies in FlipIt

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. In this paper we will focus on the non-adaptive 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.

2.3 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 [3]. 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 [4] 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. [5] is to consider a game where the moves made by the attacker are still covert but the moves made by the defender are known to the attacker. This means that the attacker can base his attacks on the defender's moves. Both the targeted and non-targeted attacks don't succeed immediately. 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. The conclusion of this paper is that the optimal strategy for the defender is moving periodically. The difference with this paper is that the delay in this paper is dependent on the number of nodes that have to be flipped in a network. This cannot be modelled with the framework in the Laszka et al. paper because the delay is chosen as an exponential distributed random variable. Another difference is that in the case of the Laszka et al. paper, the moves of the defender are considered not stealthy and so the attacker knows when the defender plays.

Other authors used the FlipIt game to apply it on a specific scenario. To be able to use the FlipIt game, modifications where required for the FlipIt model. One of the scenarios by Pham [8] 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.

Finally researchers also have investigated the behaviour of humans playing FlipIt. A. Nochenson and Grossklags [7] investigate how people really act when given temporal decisions. Reitter et al. [9] extended the work of A. Nochenson and Grossklags to include various visual presentation modalities for the available feedback during the investigation.

Hoofdstuk 3

FlipIt game with virus propagation

3.1 Introduction

3.2 FlipIt game with virus propagation

Motivatie voor het veranderen van FLipIt naar een FlipIt met viruspropagatie:

3.3 Explaining difference between FlipIt with and without virus propagation

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, the other player, 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. 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.

3.3.1 Actions of the attacker

A virus has different kind of ways of making his way through a company network. We will describe the different ways of how the virus can propagate. For start we will say that the virus or worm will be dropped on Node i and that it has k numbers of neighbours.

1. Node i is infected and will spread the virus or worm to every k neighbours and will stop infecting the neighbours in the next step

2. Node i is infected and will spread the virus or worm to every k neighbours and will keep on spreading the virus to the same neighbours in every next step

3. Node i is infected and will spread the virus to only one of the k neighbours and will stop infecting another neighbour in the next step

4. Node i is infected and will spread the virus to only one of the k neighbours and in the next step it will infect another one of the k neighbours

In the game that will be modelled in the paper we will use the settings of the first spreading method. We will not use method 2 because this kind of propagation will float the network. Because we use the settings of a mail system and contact in a mailing list the method of 3 and 4 are not used.
In the first method the node that has been infected can be again infected. If one of the neighbours infects the node again the node will infect his neighbours again. By using this spreading method we have three distinct states in which a node can be situated. An *infected state*, a *clean state* and a *spreading state*. An infected state means that the node is infected and will not spread the virus to its neighbours, a clean state means that the node is not infected on that moment and a spreading state means that the node is infected and that it will spread the virus or worm to its neighbours in the next step. We can argument this kind of propagation through a mail worm.

voorbeeld geven van zo een worm

The Attacker itself has two different ways of attacking the company network. It will only infected one node of the network and will wait for the virus to spread itself through the network. We will model two ways of attacks of an Attacker:

1. The attacker drops the virus on a random node on the network

2. The attacker drops the virus on a targeted node on the network

The attacker in this game will put a virus or worm on one of the nodes in the network. (This will happen at random.) The attacker does not know on which node the virus will be dropped. We will use this randomness because most viruses are spread via a usb stick or a shared resource. If we use this spreading method where we have a targeted attack the attacker will have more information about the network.

feit uit security rapport symantec

The attacker can choose at which rate it will drop a virus on one of the nodes on the network. The cost of dropping a virus will be the same. It will not increase. If it will increase this means that the attacker will eventually drop out of the game because it becomes to expensive.
The attacker is in control over the game if it manages to infect a subset of all the resources of the company network.

3.3.2 Actions of the defender

The attacker wants to protect all the nodes of his network. It can do so by getting back control over the resources. We will assume that the defender of the network has knowledge over his own network. Which is convenient in the real world because a company has to know how his infrastructure looks like.

The defender has two possible ways of defending its network:

1. The defender flips all the nodes of his network

2. The defender will flip a subset of the nodes of his network

The cost of flipping all the nods of the network will be greater than the cost of flipping a subset of nodes. We make this assumption because otherwise it will be beneficial for the defender to always flip all the nodes in the network.

We will also make the assumption that as a defender flips a node the node can get infected again. A flip will not be correlated to a patch but to a clean-up. Another setting of the game can be that the flip of the defender is equal to a patch and that the resource cannot be infected any more. But with this case we deviate from the flipIt game, because the attacker cannot flip the resource any more. Unless we work with different virusses every time the attacker flips. We start with the less complex game of flipping is equal to a clean-up.

waarom geen patch, wormen kunnen veranderen gaandeweg

andere mogelijkheid:

3.4 Formal definition Game

In this section we provide a formal definition of the game and the notation that we will use throughout the paper.

Playing periodically with virus propagation

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.

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 [10]. Further research can investigate the effect of relaxing this assumption.

Similarly as in [10], 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.

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 δ. 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 $\alpha_i = \dfrac{1}{\delta_i}$.

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

Let $r = \dfrac{\delta_D}{\delta_A}$. The intervals between two consecutive defender's moves have length δ_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.

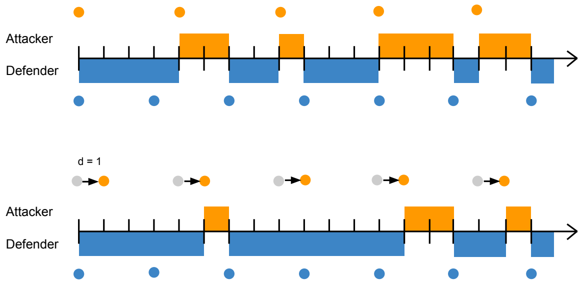
The expected period of attacker control within the interval would be r/2, without considering the delay by a virus. Therefore the benefit for the attacker, without considering the delay, can be expressed as follows:

$$\beta_A(\alpha_D, \alpha_A) = \frac{r}{2} - k_A\alpha_A = \frac{\delta_D}{2\delta_A} - k_A\alpha_A \tag{3.1}$$

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

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{r}{2} - k_D\alpha_D = 1 - \frac{\delta_D}{2\delta_A} - k_D\alpha_D \tag{3.2}$$

FIGUUR 3.1: The first FlipIt game is one without virus propagation. The second one is with virus propagation and $d = 1$. The delay is denoted with an arrow.



However, because of the delay required for virus propagation, the maximal time of control is reduced to $\delta_D - d$, see figure 3.1. 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 $\dfrac{\delta_D - d}{\delta_D}$ and this will give the attacker an average gain of $\dfrac{\delta_D - d}{2}$. 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 $\dfrac{d}{\delta_D}$. The average gain rate of the attacker can then be expressed as follows if we look at one interval of the defender:

$$\gamma_A(\alpha_D, \alpha_A) = \frac{1}{\delta_D}[\frac{\delta_D}{\delta_A} \cdot \frac{\delta_D - d}{\delta_D} \cdot \frac{\delta_D - d}{2} + \frac{\delta_D}{\delta_A} \cdot \frac{d}{\delta_D} \cdot 0] \tag{3.3}$$

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

$$\beta_A(\alpha_D, \alpha_A) = \frac{(\delta_D - d)^2}{2 \cdot \delta_D\delta_A} - k_A\alpha_A \tag{3.4}$$

$$\beta_A(\alpha_D, \alpha_A) = \frac{\delta_D}{2 \cdot \delta_A} - k_A\alpha_A - (\frac{d^2}{2 \cdot \delta_A\delta_D} - \frac{d}{\delta_A}) \tag{3.5}$$

The benefit of the defender is expressed as follows:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{(\delta_D - d)^2}{2 \cdot \delta_D\delta_A} - k_D\alpha_D \tag{3.6}$$

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 δ_A. Consider a given attackers 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:

$$\beta_D(\alpha_D, \alpha_A) = \frac{1}{2r} - k_D\alpha_D = \frac{\delta_A}{2\delta_D} - k_D\alpha_D \tag{3.7}$$

$$\beta_A(\alpha_D, \alpha_A) = 1 - \frac{1}{2r} - k_A\alpha_A = 1 - \frac{\delta_A}{2\delta_D} - k_A\alpha_A \tag{3.8}$$

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: $d + \delta_A \leq \delta_D$

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 $\delta_A + d \leq \delta_D$. 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 3.2 cases (1) and (2). This means that the defender will keep the control over the resource in the next interval over a period of the delay, namely 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 $\frac{(d + \frac{\delta_A}{2})}{\delta_A}$

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:

$$\gamma_D(\alpha_D, \alpha_A) = \frac{\delta_A}{\delta_D} \cdot \frac{(d + \frac{\delta_A}{2})}{\delta_A} \tag{3.9}$$

$$\gamma_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} \tag{3.10}$$

This yields in the following benefit formula:

$$\beta_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D\alpha_D \tag{3.11}$$

The benefit for the attacker will be as follows:

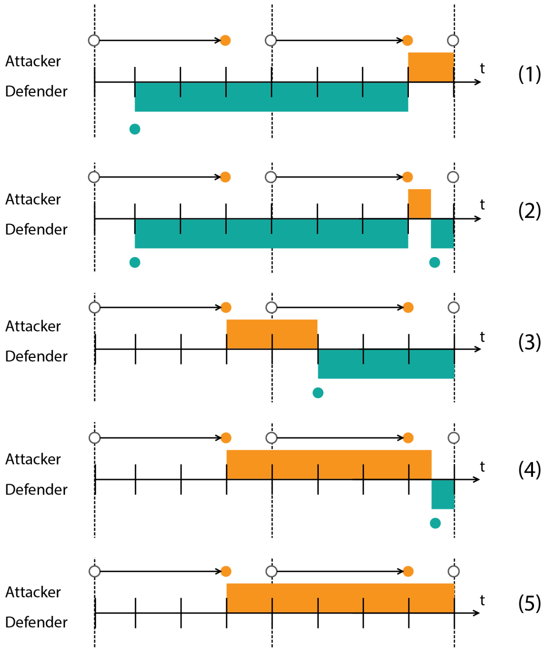
$$\beta_A(\alpha_D, \alpha_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{d}{\delta_D} - k_A\alpha_A \tag{3.12}$$

It is crucial that δ_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 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

Figuur 3.2: Case 2 where d + deltaA < deltaD



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:

$$\frac{1}{\delta_A} \cdot \frac{d-(\delta_D-\delta_A)}{\delta_D} \cdot \frac{d-(\delta_D-\delta_A)}{\delta_D} \tag{3.13}$$

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

$$\gamma_D(\alpha_D, \alpha_A) = \frac{\delta_A}{\delta_D} \cdot \frac{(d + \frac{\delta_A}{2})}{\delta_A} - \frac{(d-(\delta_D-\delta_A))^2}{2\delta_D\delta_A} \tag{3.14}$$

$$\gamma_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - \frac{(d-(\delta_D-\delta_A))^2}{2\delta_D\delta_A} \tag{3.15}$$

This yields in the following benefit formula:

$$\beta_D(\alpha_D, \alpha_A) = \frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D\alpha_D - \frac{(d-(\delta_D-\delta_A))^2}{2\delta_D\delta_A} \tag{3.16}$$

The benefit for the attacker will be as follows:

$$\beta_A(\alpha_D, \alpha_A) = 1 - \frac{\delta_A}{2\delta_D} - \frac{d}{\delta_D} - k_A\alpha_A + \frac{(d - (\delta_D - \delta_A))^2}{2\delta_D\delta_A} \tag{3.17}$$

3.5 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 (α_0^*, α_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 opt0(α_1) the set of values (rates of play α_0) that optimize the benefit of the defender for a fixed rate of play α_1 of the attacker. Similarly, we denote by opt1(α_0) the set of values (rates of play α_1) that optimize the benefit of the attacker for a fixed rate of play α_0 of the defender. The following theorem specifies Nash equilibria for the periodic game and is proven in Appendix A.

Nash equilibria are points whith the property that neither player benefits by deviating in isolaition form equilibrium. We can compute Nash Equilibria for the periodic game as an intersection points of curvest opt0 and opt1. To determine opt0(a0) we need to compute the derivate of $\beta_0(\alpha_0, \alpha_1)$ for a fixed α_1. We consider two cases:

—–

3.5.1 case: $\delta_D \leq \delta_A$ or $\alpha_D \geq \alpha_A$

Since

$$\frac{\partial\beta_D(\alpha_D, \alpha_A)}{\partial\alpha_D} = \frac{\partial[1 - \frac{(\delta_D - d)^2}{2 \cdot \delta_D\delta_A]} - k_D\alpha_D}{\partial\alpha_D} \tag{3.18}$$

If we change every $\delta = \frac{1}{\alpha}$ in formule 3.18 we get the next formula that is easier to take the derivate of:

$$\frac{\partial\beta_D(\alpha_D, \alpha_A)}{\partial\alpha_D} = \frac{\partial[1 - \frac{(1/\alpha_D - d)^2 \cdot \alpha_D\alpha_A}{2} - k_D\alpha_D]}{\partial\alpha_D} \tag{3.19}$$

Which gives the following result:

$$\frac{\partial\beta_D(\alpha_D, \alpha_A)}{\partial\alpha_D} = \alpha_A(\frac{1}{\alpha_D^2} - \frac{d^2}{2}) - k_D \tag{3.20}$$

it follows that if equal to zero $\alpha_0 = \sqrt{\dfrac{\alpha_A}{2k_D + d^2\alpha_A}}$

3.5.2 case 2: $\delta_D \geq \delta_A$ or $\alpha_D \leq \alpha_A$

case A

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{\partial[\frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D\alpha_D]}{\partial\alpha_D} \tag{3.21}$$

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{\partial[\frac{\alpha_D}{2\alpha_A} + d\alpha_D - k_D\alpha_D]}{\partial\alpha_D} \tag{3.22}$$

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{1}{2\delta_A} + d - k_D \tag{3.23}$$

Dus increasing als $\alpha_A < \dfrac{1}{2(k_D - d)}$ and decreasing if $\alpha_A > \dfrac{1}{2(k_D - d)}$

Case b

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{\partial[\frac{\delta_A}{2\delta_D} + \frac{d}{\delta_D} - k_D\alpha_D - \frac{(d-(\delta_D-\delta_A))^2}{2\delta_D\delta_A}]}{\partial\alpha_D} \tag{3.24}$$

Uitgewerkt geeft de benefit formule de volgende formule omgezet naar alpha

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{\partial[1 - \frac{\alpha_A}{2\alpha_D} - k_D\alpha_D + \frac{d\alpha_A}{2} + \frac{d\alpha_D}{2} - \frac{d^2\alpha_A\alpha_D}{2}]}{\partial\alpha_D} \tag{3.25}$$

$$\frac{\partial\beta_D(\alpha_D,\alpha_A)}{\partial\alpha_D} = \frac{\alpha_A}{2\alpha_D^2} - k_D + \frac{d}{2} - \frac{d^2\alpha_A}{2} \tag{3.26}$$

Als ge dat gelijk stelt aan 0 krijgt ge:

$$\alpha_D = \sqrt{\frac{\alpha_A}{2k_D - d + d^2\alpha_A}} \tag{3.27}$$

Hoofdstuk 4

Virus Propagation

4.1 Methods of virus propagation

4.2 REpresenting Virus Propagation in a network

4.3 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 toegedaan.

– 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

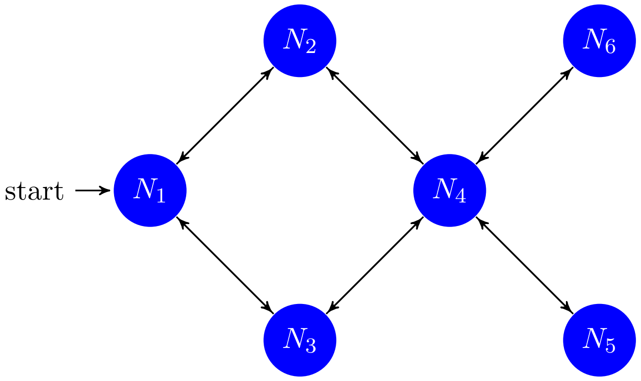
Stuxnetworm is a very famous worm. Initially this worm spread via infected USB sticks and from then it could spread through the Internet to other computers. The purpose of the Stuxnetworm was broken to run the centrifuges in nuclear reactors. Many reactors have been infected. From the standpoint of the defender, it is very important to respond as quickly as possible so that the worm can not spread quickly.

4.4 Representing Virus Propagation in a network

4.5 methode met matrixen

source: `http://en.wikipedia.org/wiki/Adjacency_matrix`
We model the network through an undirected Graph $G =< V, E >$ 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 $[A]$:

$$\begin{array}{c|cccccc} & N_1 & N_2 & N_3 & N_4 & N_5 & N_6 \\ \hline N_1 & 0 & 1 & 1 & 0 & 0 & 0 \\ N_2 & 1 & 0 & 0 & 1 & 0 & 0 \\ N_3 & 1 & 0 & 0 & 1 & 0 & 0 \\ N_4 & 0 & 1 & 1 & 0 & 1 & 1 \\ N_5 & 0 & 0 & 0 & 1 & 0 & 0 \\ N_6 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

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]*

$$\begin{array}{c|cccccc} & N_1 & N_2 & N_3 & N_4 & N_5 & N_6 \\ \hline N_1 & 2 & 0 & 0 & 2 & 0 & 0 \\ N_2 & 0 & 2 & 2 & 0 & 1 & 1 \\ N_3 & 0 & 2 & 2 & 0 & 1 & 1 \\ N_4 & 2 & 0 & 0 & 4 & 0 & 0 \\ N_5 & 0 & 1 & 1 & 0 & 1 & 1 \\ N_6 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

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]*

$$\begin{array}{c|cccccc} & N_1 & N_2 & N_3 & N_4 & N_5 & N_6 \\ \hline N_1 & 0 & 4 & 4 & 0 & 2 & 2 \\ N_2 & 4 & 0 & 0 & 6 & 0 & 0 \\ N_3 & 4 & 0 & 0 & 6 & 0 & 0 \\ N_4 & 0 & 6 & 6 & 0 & 4 & 4 \\ N_5 & 2 & 0 & 0 & 4 & 0 & 0 \\ N_6 & 2 & 0 & 0 & 4 & 0 & 0 \end{array}$$

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 $G =< V, E >$

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 δ_0 of defender and δ_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 & contoller=attacker )
d:= d + 1
M = M x A
T = T + M
G = X x T
```

Hoofdstuk 5

Conclusion

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

5.1 trala

Bibliografie

[1] Advanced persistant threats (apt).

[2] Gametheory, 2004.

[3] 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.

[4] A. Laszka, B. Johnson, and J. Grossklags. Mitigating covert compromises. *iets*, 8289:319–332, 2013.

[5] A. Laszka, B. Johnson, and J. Grossklags. Mitigation of targeted and non-targeted covert attacks as a timing game. 8252:175–191, 2013.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. van Dijk, A. Juels, A. Oprea, and R. Rivest. Flipit: The game of ”stealthy takeover”. *Journal of Cryptology*, 26(4):655–713, 2013.

KU Leuven Faculteit Ingenieurswetenschappen 2014 – 2015

Fiche masterproef

Student: Sophie Marien

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

Engelse titel: Beste masterproef ooit al geschreven

UDC: 621.3

Korte inhoud:

Hier komt een heel bondig abstract van hooguit 500 woorden. LaTeX commando's mogen hier gebruikt worden. Blanco lijnen (of het commando `\par`) zijn wel niet toegelaten!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

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

Promotor: Prof. dr. ir. Tom Holvoet

Assessoren: Ir. W. Eetveel
W. Eetrest

Begeleider: Ir. Jonathan Merlevede, Ir. Kristof Coninx