KU LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN

Gametheory and Cybersecurity: a study FlipIt and multiple resources

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. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my wive and the rest of my family.

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.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lijst van figuren en tabellen

Lijst van figuren

Lijst van tabellen

List of Abbreviations and Symbols

Abbreviations

LoG	Laplacian-of-Gaussian
MSE	Mean Square error
PSNR	Peak Signal-to-Noise ratio

Hoofdstuk 1

Introduction

The first contains a general introduction to the work. The goals are defined and the modus operandi is explained.

bib referenties in orde brengen

1.1 Introduction

Security is an important asset in Computer Science. Defending a network of a company is not an easy job. To prevent intruders it can make use of firewalls, routers, IDS systems, virus scans, and other defence mechanisms. Unfortunately technology is growing fast and attacks are getting more sophisticated and the causes of these attacks can be very different. Companies are often the victim of targeted attacks. In a security report of 2014, , states that 80% of the companies are the victims of targeted attacks. Many companies don't see themselves as a target, but sometimes they might be collateral, the target on the way to the real target. This means that everybody can be a target. Corporate networks should continuously defend themselves against outside invaders such as viruses and worms. By doing so the network administrator can keep the network as malware-free as possible. If there is an intruder managed to penetrate the network then the network manager this intruder trying to get out as quickly as possible. This is not always easy. Especially when the intruders secretly sneak and then spread rapidly. In this paper we will work further on the work made by Marten van Dijk, Ari Juels, Alina Oprea and Ronals L. Rivest who wrote a report on the Game FlipIt. FlipIt is a the game of " Stealthy Takeovers". It models a game by means of two players, the attacker and the defender. Both can gain control over a single shared resource by flipping it. The most important property of the game is that the flipping happens stealthy. This means that the players have no clue about when the other player moves. The goal of the game is to maximise the time the player controls the resource minus the average cost of the flipping.

iets tussen nog

verwijzing naar report

verwijzing naar FlipIT

1.1.1 Motivation of the game

1.1.2 Contributions and results

1.1.3 Conclusions

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.

1.2 introduction number 2

security report van pwc

withepaper toe-voegen

(We live in an era) In this era where digitalization becomes prominent in every aspect of our lives, where technology is growing fast and where business are always under attack, security becomes an issue of increasing complexity. Since 2009, the number of reported security attacks has increased 66%, year over year. . These numbers only represent the attacks that are detected. In 2014 117,339 attacks where coming in daily. Many of those attacks have a different cause. Some of them can be benign, others can be harmful. Many companies are unaware of all the attacks. Some of them think that they are not a target, but they might be a target on the way to a real target. Recently there where some high profiled targeted attacks which have been revealed. (Belgacom). Targeted attacks are ... The *Kill Chain* is a concept by Lockheed Martin Corporation, explained in the whitepaper . It explains the different phases of a typical attack from the view of an attacker. It also outlines the typical attacker activities on the right. This model is very useful to define the different moments of the life cycle of an attack and when a company should act to defend itself. In this paper we would like to prevent the viruses of spreading into the network system of a company. This means that we have to act in phase Installation, Command and Control and Action on Objectives of the kill chain.

Security is an important asset in Computer Science. Defending a network of a company is not an easy job. Malicious people will try to To prevent intruders it can make use of firewalls, routers, IDS systems, virus scans, and other defence mechanisms. Unfortunately technology is growing fast and attacks are getting more sophisticated and the causes of these attacks can be very different. Companies are often the victim of targeted attacks. In a security report of 2014, , states that 80% of the companies are the victims of targeted attacks. Many companies don't see themselves as a target, but sometimes they might be collateral, the target on the way to the real target. This means that everybody can be a target. Corporate networks should continuously defend themselves against outside invaders and targeted attacks. Researchers have already investigated the situations through the FlipIt game in which a system is continuously compromised by an attacker through targeted attacks. FlipIt is a the game of " Stealthy Takeovers". It models a game by means of two players, the attacker and the defender. Both can gain control over a single shared resource by flipping it. The most important property of the game is that the flipping happens stealthy. This means that the players have no clue about when the other player moves and has control over the shared resource. The goal of the game is to maximise the time the player controls the resource minus the average cost of the number of flipping. In this paper we model a company network through multiple shared resources and a flip from the attacker that drops a virus that will spread itself autonomously. We show that ...

iets tussen nog

verwijzing naar report

Hoofdstuk 2

Intoduction to GameTheory

In the following paragraph an introduction to game theory is given based on the work of [**?**] and [**?**]. For a more detailed and full introduction to game theory, the reader is referred to [**?**].

2.1 Intro 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. For this reason it is very important for economics and also for 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 to tell you how much an agent likes or dislikes the states of the world.
It also explains the impact of uncertainty. When an agent is uncertain about a distribution of outcomes, his utility will describe the expected value of the utility function with respect to the probability of the distribution of the outcomes. For example: with 0.7 probability it will be 7 degrees outside and with 0.3 probability it will be 10 degrees. The agent can have a different opinion about that distribution versus another distribution. (). uitleggen aan de hand van een voorbeeld
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 whose actions can affect each other 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.

There are two standard representations for games. The first one is the Normal Form. The second one is the Extensive Form.

In the following lists 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.

Actions: actions are what the player can do.

Outcomes:

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.

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

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

2.2 Virusses

Many network security threats today are spread over the Internet. The most common include:

Viruses, worms, and Trojan horses Spyware and adware Zero-day attacks, also called zero-hour attacks Hacker attacks Denial of service attacks Data interception and theft Identity theft

Computer virus through mail. Though virus spreading through email is an old technique, it is still effective and is widely used by current viruses and worms. Sending viruses through email has some advantages that are attractive to virus writers: Sending viruses through email does not require any security holes in computer operating systems or software. Almost everyone who uses computers uses email service. A large number of users have little knowledge of email viruses and trust most email they receive, especially email from their friends [28][29]. Email are private properties like post office letters. Thus correspondent laws or policies are required to permit checking email content for detecting viruses before end users receive email [18].

Send a email with malicious attachment. Only again infected if attachment again opened. Thus this is the action of attacking every neighbour node + also can attack again the node where the virus was coming from. There are also email viruses were the malicious program is hidden in the txt and the attachment does not need to be opened.

2.2.1 Malware

Relevant researches:

- How Viruses and worm can be detected. Difference between UDP en TCP worm propagation. Difference for the propagation speed.

Purpose thesis: model a worm propagation with adaptations of Flip-It. Flip-It cannot address the evaluation of individual nodes. Flip-It with multiple resources has not addressed the fact that a virus does not need to compromise the whole network. A subpart of the network can already cause problems for the company. Data leackages. So the virus can propagate

=======

2.3 Conclusion

The final section of the chapter gives an overview of the important results of this chapter. This implies that the introductory chapter and the concluding chapter don't need a conclusion.

Hoofdstuk 3

The FlipIt game

In this chapter, we introduce the game FlipIt [**?**]. FlipIt is a game introduced by .. et al. First we explain the framework of FlipIt and it's important results. In the next section the formulas and assumptions are made that will be used throughout the paper. 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 it's notations.

3.1 The FlipIt game

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 rest of the paper we will call the two players the attacker, denoted by the subscript A and the defender, denoted by subscript D.

The game begins at $t = 0$ and continuous indefinitely ($t \to \infty$). The time in the game can be viewed as being continuous, but a discrete time can also be viewed. To get control over the resource, the players i 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. By adding a cost, it will prevent players to move to frequently.

The unique feature of FlipIt is that every move will happen in a stealthy way, meaning that the player has no clue that (his adversary) the other player has flipped the resource. For instance, the defender will not find out if the resource has already been compromised by the attacker, but he can only potentially know it after 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 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 defender. If the defender 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 and a cost is involved.

The state of the resource is denoted as a time independent variable $C = C_i(t)$. $C_D(t)$ is either 1 if the game is under control by the defender and 0 if the game is under control by the attacker. For $C^A(t)$ it is visa versa, $C^A(t) = 1 - C^D(t)$.

The game starts with defender being in control of the game, $C_D(0) = 1$.

The players receive a benefit equal to the time of units that 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:

$$G_i(t) = integraal[0][t]C_i(x)dx. \tag{3.1}$$

The average gain of player i is defined as:

$$\gamma_i(t) = G_i(t)/t. \tag{3.2}$$

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

$$\beta_i(t) = \gamma_i(t) - k_i\alpha_i. \tag{3.3}$$

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.

Because the players move in a stealthy way, there are different types of feedback that a player can get while moving:

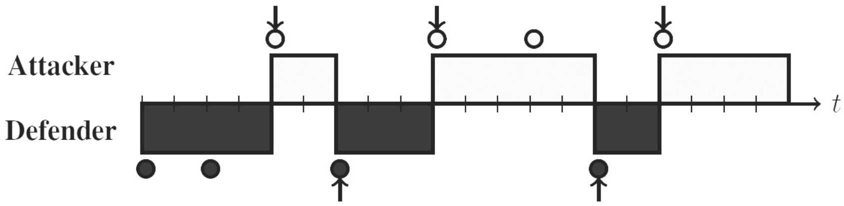
- Non-adaptive (NA): The player does not receive any feedback during the game while flipping.
- Last move (LM): When a player flips it will find out the exact time that the opponent played the last time.
- Full History (FH): When a player flips it will find out the whole history of the opponents move.

The game can be extended by the amount of information that a player receives. It can also be possible for a player to get information at the start of the game. Both interesting cases are:

- Rate-of-play (RP): The player finds out the exact rate of play of the opponent.
- Knowledge-of-strategy (KS): The player finds out the complete information of the strategy that the opponent is playing.

In our assumption the strategy of both players will be non-adaptive. None of the players has information of the strategy of the opponent.

verwijzen naar de figuur 3.1



FIGUUR 3.1: The FlipIt game where both players are playing periodically

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

TABEL 3.1: Classes of strategies in FlipIt

3.1.1 Strategies

In this subsection we go through the strategies used in FlipIt and the most important results.

There are two different kinds of strategies, the *non-adaptive strategies* and the *renewal strategies*. If there is no need for feedback for both of the players, we say that we have a non-adaptive strategy. Because the player does not receive any feedback during the game it will play in the same manner against every opponent. They are not dependent on the opponents movements. This means that they can already generate the time sequence for all the moves in advance. But they can depend on some randomness because the non-adaptive strategies can be randomised. In this paper we will focus in the beginning on the non-adaptive strategies. Reasons behind this is that a player (defender or attacker) rarely knows what the strategies are of his opponent. [If the attacker wants to move stealthily, it might have limited attack options FLIPTHEM].

nog redenen zoeken

A renewal strategy is a non-adaptive strategy where the time intervals between two consecutive moves are generated by a renewal process.

Periodic

Non-Arithmetic Renewal

Exponential

Hoofdstuk 4

FlipIt game with virus propagation

4.0.2 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.

4.0.3 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:

4.0.4 Strategies of both players

We explained what the actions of each player are.

4.1 Formal definition Game

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

Players There are two players in the game, one is the defender and the other one is the attacker. They are respectively identified by 0 and 1.

Time The game starts at $t = 0$ and continuous indefinitely as $t \to \infty$. The game is a continuous game.

Graph We represent the company network as a Graph $G =< V, E >$. G is an ordered pair where V denotes the set of resources or nodes in the network and E denotes the set of connections or links, which are a two-element subset of V. We use the notations resources and nodes interleaving in this paper.
We have N resources in the network. $N \in$. This means we can denote the resources by: aanvullen

$$V \in V_0, V_1, V_2, ..., V_N$$

The set E of connections indicates if there is a link between two resources. We see the links as bidirectional so the total graph is undirected. If there is a link between resource V_n and V_{n+1} then there is also a link between V_{n+1} and V_n.

Game State There is also a time-dependent variable that represents the state of the game. $C = C(t)$ is either 0 if the game is under control by the defender and 1 if the Game is under control by the attacker.
We start at $t = 0$ with the defender who has control over the game. We do this because we assume that the defender will only put the network online without having a virus or worm in it. The Attacker can gain control over the game when it compromises a subset s of the resources. The subset s is a minimum of 1 resource and a maximum of all the resources N.
We can also define the state of each resource by C_N^A and C_N^D. If $C_N^A = 1$ then this means that the attacker has control over the resource, and 0 otherwise. For C_N^D it is visa versa, $C_N^D = 1 - C_N^A$. deze variabele nodig ja of nee ? JA

Moves Both players can make a move in the game. Moves done in a finite numbers of time in any finite time interval. Both players can play at any time they want, they can also play at the same time. If this happens the one that has control over the resource will keep having control over the resource. This makes the game fully symmetric . The sequence of move times are denoted by the following infinite sequence: beter uitleggen

$$t = t_1, t_2, t_3, ..$$

Two move times can be the same because we allow players to move at the same time. We can also denote the infinite sequence of times when player i moves. We write this as :

$$t = t_{i,1}, t_{i,2}, t_{i,3}, .. \text{ with } i \in \{0, 1\}$$

The sequences t_1 and t_0 are disjoint subsets of the sequent t. We can also denote who made the kth move by defining a sequence p that denotes the sequence of who played:

$$p = p_1, p_2, p_3, .. \text{ with } p_k \in \{0, 1\}$$

Number of moves $n_i(t$ denotes the number of moves made by player i up to and including time t. This means that

$$n(t) = n_1(t) + n_0(t)$$

is the sum of the number of moves made by the defender and the attacker up to and including time t.

Average move rate We denote $\alpha_i(t)$ as the average move rate by player i:

$$\alpha_i(t) = n_i(t)/t \text{ with } t > 0 \text{ and } i \in \{0, 1\}$$

Period We can also define the period in terms of the average move rate:

$$\delta_i = 1/\alpha_i$$

Who played last We know who played last by taking the modulo with the period. Z_i represents the time since the last flip of player i. We can also denote the time since the last flip of player i on resource r by Z_i^N. For a non adaptive game, period deterministic: At time $t = n$ is $Z_i = nmodi$. .

er kan nog steeds tegelijk geflipt zijn maar dan hebben ze wel geflipt

Cost The cost is an important property of the game. In FlipIt for every player the cost of a move is denoted by k_i. These costs can be very different for every player. In this game we denote the players flipping cost for resource V_N by $c_i^{V_N}$.
For the defender the cost will be either the cost of flipping every resource or the cost of flipping a subgroup of the resources.
For the attacker the cost will be the cost of dropping a virus on a node. The spreading of the virus will not imply an extra cost.

ain van een
urce, moet
verschil-
e resources

Utility In FlipIt the Gain definition is the utility function. The Gain denotes the total time a player i has gained control over a resource. The Gain G_i denotes players i total gain of a game, which is the total time the player has gained control over a subset of resources thus controlling the game. This is denoted by the following:

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

If we sum up the total Gain of the attacker and the defender we end up with the time:

$$G_1(t) + G_0(t) = t$$

Average gain rate The average gain rate for player i is defined as

$$\gamma_i(t) = G_i(t)/t$$

4.1.1 Formal definition

Graph Matrix We represent the graph of the network through a matrix $A = |V| \times |V|$. The (i,j)-entry of the matrix A will have a 1 if there is a connection between node V_i and node V_j. If we are working with an undirected graph the matrix will be symmetric.

Attack Vector We denote $X = 1 \times |V|$ as the attack vector. It will be a vector with only zeros. The attacker will place a virus on a node V. This will be denoted by the Vth entry in the vector that is changed by a 1.

Reset vector The reset vector will make sure that the right entries in the matrix become zero. If the defender flips every node every time it flips then the attack vector will be 0.

Cummulative Matrix This matrix will keep record of the propagation of the virus through the network.

State Matrix The State matrix $T(t) = 1 \times |V|$ will keep at every time t the state of the game and denote which node at time t is infected with the virus. At time $t = 0$ the State Matrix will be the null matrix.

De eerste infectie is de attack vector * Graph matrix .

4.2 Conclusion

The final section of the chapter gives an overview of the important results of this chapter. This implies that the introductory chapter and the concluding chapter don't need a conclusion.

4.3 Extensions on FlipIt

There a various possible ways to extend FlipIt. For instance Laszka et al. extended the basic FlipIt game to multiple resources. The incentive is that for compromising a system in a real case it needs more than just taking over one resource. An example is gaining access to a system and breaking the password. The model is called FlipThem [**?**]. Two ways of flipping the resources are used: 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. The difference with FlipThem and this paper is that we introduce a Graph Model in the beginning.

citatie needed voor Are We Compromised?

Another extension on FlipIt is done by Pham[**?**] []. Beside the action Flip their is another action Test. The basic idea is to test with an extra action if the resource has been compromised or not. This action involves also an extra cost. This model is useful if somebody wants to know for example if his password has been compromised or wants to assess the periodic security of a system. In [**?**] [**?**] Laszka et al. they also consider non targeted attacks by non-strategic players and .

verder aanvullen

Hoofdstuk 5

Intoduction to GameTheory

5.1 Gametheory

Game theory is the study between self interested agents and the actions they perform. self interested agents means not that they want to harm each other or that they only care about themselves. Agent has a description of the world he likes. A utility function is a mapping of the world that the agent likes to real numbers.

5.2 Literatuurstudie

Difference with FlipThem: sub part of nodes for control and strategy difference: dependant of grade of the nodes, instead of just periodic. - Literatuurstudie - Flipit - Game Motivation - Formal definition

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'.

5.2.1 What can be done in further research

- Looking for the dynamics of the spread of the virus/worm limited by the bandwidth of the network links, BPG routing failure with high volume scan traffic

5.3 Conclusion

5.4 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.

5.5 ..

Flip-it. Some authors have written other papers about flipit. One of them is the [Game theoretic approach for cost-benefit analysis of malware proliferation prevention].

Company networks are targeted. It costs a lot. They want to defend their company networks. No loss of data, integrity and confidentiality. Many ways to attack a company. Viruses, Trojans, worms, DOS, .. Hard to protect against every attacker.

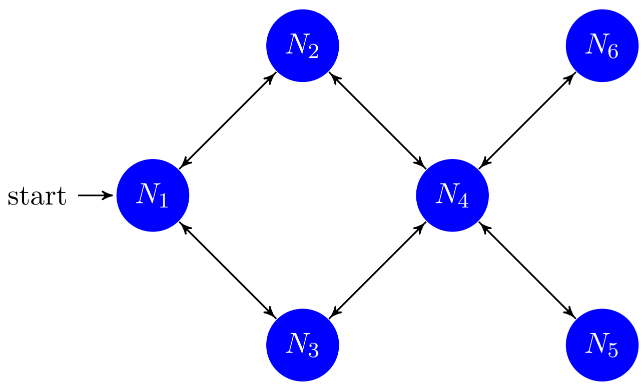
Hoofdstuk 6

Intoduction to GameTheory

6.1 Write down the settings of the game

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 $[A]$:

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

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]*

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

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]*

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

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 $G =< V, E >$

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 7

Intoduction to GameTheory

In the following paragraph an introduction to game theory is given based on the work of leyton2008essentials and Coursera. For a more detailed and full introduction to game theory, the reader is referred to leyton2008essentials.

7.1 Intro 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. For this reason it is very important for economics and also for 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 to tell you how much an agent likes or dislikes the states of the world.
It also explains the impact of uncertainty. When an agent is uncertain about a distribution of outcomes, his utility will describe the expected value of the utility function with respect to the probability of the distribution of the outcomes. For example: with 0.7 probability it will be 7 degrees outside and 0.3 probability it will be 10 degrees. The agent can have a different opinion about that distribution versus another distribution. ().

uitleggen aan de hand van een voorbeeld

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 whose actions can affect each other 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

players rationeel en max outcomes

game theory.

There are two standard representations for games. The first one is the Normal Form. The second one is the Extensive Form.

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.

Actions: actions are what the player can do.

Outcomes:

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.

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

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

Best response ook uitleggen?

Voorbeeld ook nog uitleggen?

One of the solution concepts in Game Theory for non-cooperative games is a Nash Equilibrium. 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 payoff 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.

7.2 Virusses

Stealth Regin's developers put considerable effort into making it highly inconspicuous. Its low key nature means it can potentially be used in espionage campaigns lasting several years. Even when its presence is detected, it is very difficult to ascertain what it is doing. Symantec was only able to analyze the payloads after it decrypted sample files.

It has several "stealth" features. These include anti-forensics capabilities, a custom-built encrypted virtual file system (EVFS), and alternative encryption in the form of a variant of RC5, which isn't commonly used. Regin uses multiple sophisticated means to covertly communicate with the attacker including via ICMP/ping, embedding commands in HTTP cookies, and custom TCP and UDP protocols Ways of defending a network:

- Self-defending networks: The next generation of network security

- Honeynet games: a game theoretic approach to defending network monitors

Many network security threats today are spread over the Internet. The most common include:

Viruses, worms, and Trojan horses Spyware and adware Zero-day attacks, also called zero-hour attacks Hacker attacks Denial of service attacks Data interception and theft Identity theft

Computer virus through mail. Though virus spreading through email is an old technique, it is still effective and is widely used by current viruses and worms. Sending viruses through email has some advantages that are attractive to virus writers: Sending viruses through email does not require any security holes in computer operating systems or software. Almost everyone who uses computers uses email service. A large number of users have little knowledge of email viruses and trust most email they receive, especially email from their friends [28][29]. Email are private properties like post office letters. Thus correspondent laws or policies are required to permit checking email content for detecting viruses before end users receive email [18].

Send a email with malicious attachment. Only again infected if attachment again opened. Thus this is the action of attacking every neighbour node + also can attack again the node where the virus was coming from. There are also email viruses were the malicious program is hidden in the txt and the attachment does not need to be opened.

Spy vs Spy: Aldrich Ames was a CIA Counter-Intelligence officer. He was also a spy feeding valuable intelligence to the Soviets and compromising US intelligence operations in the Soviet Union. He operated for 9 years before the CIA recognized that they had a spy and began an investigation and determined that he was the leak. This strategic situation is the same one faced by computer networks, drug cartels, intelligence agencies and guerrilla networks.

All such organisations have a reasonable expectation that trusted personal/systems will eventually be recruited/captured by enemy organisations. Therefore such organisations must consume valuable resources to discover such betrayals and thereby regain secrecy. The question is then given the possible threats how often and at what cost should they spend resources on investigations/spy hunts/virus scans. This is where flipIt comes in.

FLIPIT: The Game of ”Stealthy Takeover:” FlipIt was created to model these sorts of strategic situations and to study the best courses of action. Specifically flipIt was motivated by the recent interest in and success of Advanced Persistent Threats, or APT.

The basic idea is that given the current experience that perfect protection of trusted resources is unattainable, lets think about how we can optimally manage compromises of the our most trusted systems.

Rules

Two players, player X (blue) and player Y (red) attempt to maintain control over a shared resource. At anytime in the game each player is allowed to play ’flip’. The only way a player can learn the state of the game (who is in control) is when they play flip. If a player is in control of the resource and they play flip they remain in control of the resource. If a player is not in control of the resource and they play flip they gain control of the resource. Players gain points for the length of time they control the resource. Players lose points every time they play flip. This reflects the situation that the CIA is placed in with regard to moles/enemy spies. They don't know if they have been compromised. They can perform an investigation and determine if they have been compromised, also catching the spy in the act, but this action is very expensive. That is, the CIA has to trade off between remaining"mole free" (a good) and investigations (an expense).

Winning: How do you win a fair game of flipIt against intelligent adaptive human adversaries? I'm not sure.

In the real world what is the best move given that the other players can secretly capture/corrupt your most trusted personal/systems? Rives suggests in his talk that you: Be prepared to deal with repeated total failure (loss of control). Play fast! Aim to make opponent drop out! Arrange game so that your moves cost much less than your opponent's!

7.2.1 Malware

Relevant researches:

- How Viruses and worm can be detected. Difference between UDP en TCP worm propagation

7.3 Conclusion

The final section of the chapter gives an overview of the important results of this chapter. This implies that the introductory chapter and the concluding chapter don't need a conclusion.

Hoofdstuk 8

FlipIt with a virus propagation

resources vs nodes

8.1 Introduction

This section gives a formal definition of the FlipIt game with a virus propagation. First we derive a formula for a FlipIt game without a virus. After that we introduce a modification to this formula to achieve an adapted formula for a FlipIt game with a virus propagation.

8.1.1 FlipIt vs FlipIt with virus propagation

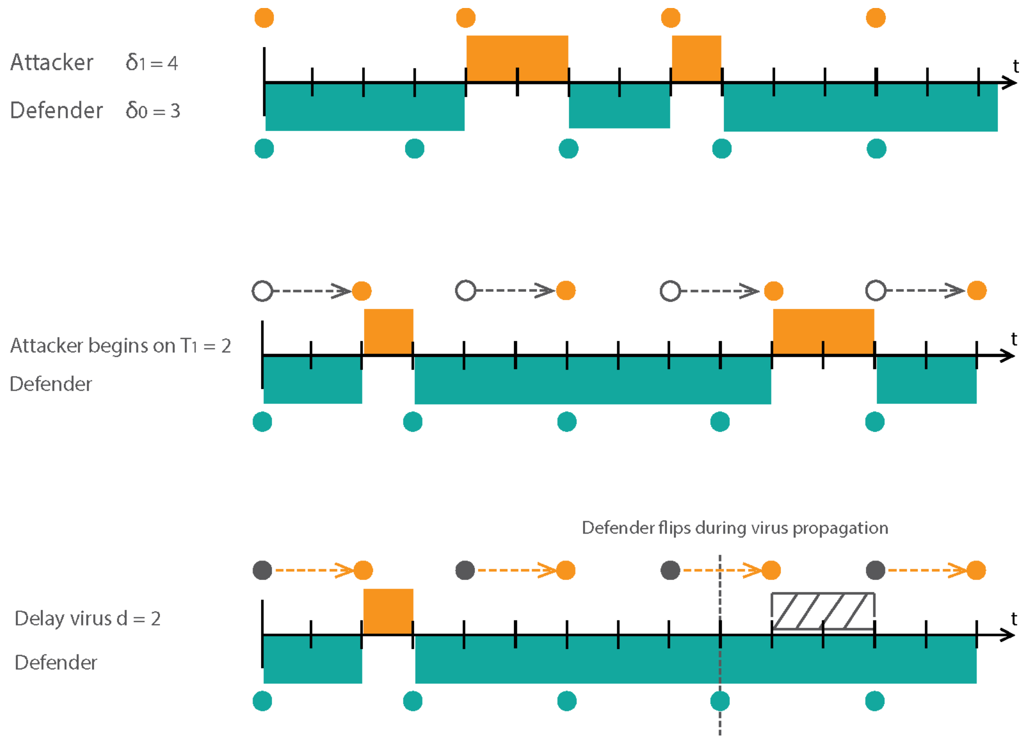
Section [] explained the FlipIt game. This section will expands the FlipIt game to a FlipIt game with virus propagation. A FlipIt game with virus propagation is a game with multiple nodes where the attacker will drop a virus on one of the nodes. The virus will then spread itself to the neighbour nodes trying to infect every node. The attacker will only gain control over the whole network -or the game in general- when it has infected all the nodes.

After dropping a virus on the first resource, it takes a while for the virus to infect the entire network. The time that it takes for the virus to infect every resource will be denoted as parameter d. If we want to measure how long it takes for the virus to infect all the resources, we have to calculate the shortest path from the first infected node to the farthest node. This can be measured by a method that we will explain in section []. Using this parameter, the gain of a FlipIt game will be calculated in section [].
The gain of a player is defined as the total amount of time that a player has owned the resource since the start of the game. In the context of a FlipIt game with virus propagation, the whole network is seen as one resource. This resource is owned by the attacked if he has control over the entire network. If the defender owns at least one node, he owns the resource.

uitleggen waarom we een gewone flippit kunnen nemen met 1 resource

a virus propagation

Figuur 8.1: Difference in a FlipIt game between delay caused by a virus and a phase bigger than zero for the Attacker



8.1.2 define formula

deze sectie herschrijven

There is a definition given for the gain of a player i by the writers of the paper FlipIt, but we want to add the property of a virus propagation to the game, hence parameter d, so we are trying to find another formula that defines a game by counting the amount of time one of the players has control.

we hebben een periodisch spel, simpelste spel.

ergens vermelden waarom we deze case bespreken en niet waarbij de aanvaller niet periodisch speelt

zeggen dat we eerst de functie voorstellen van de normale FlipIt

First a list of notations that will be used throughout the formal definition (see figure 9.1 for a graphic representation of some of the notations):

δ_0: This is the period of the defender. This denotes the length of the interval between two consecutive moves of the defender.

δ_1: This is the period of the attacker. This denotes the length of the interval between two consecutive moves of the attacker.

T_0: This denotes the phase of the defender that was random and uniform chosen over the interval $[0,\delta_0]$.

T_1: This denotes the phase of the attacker that was chosen uniformly at random in interval $[0,\delta_1]$.

Unit of control: Defined as the period between gaining control and losing control over the resource.

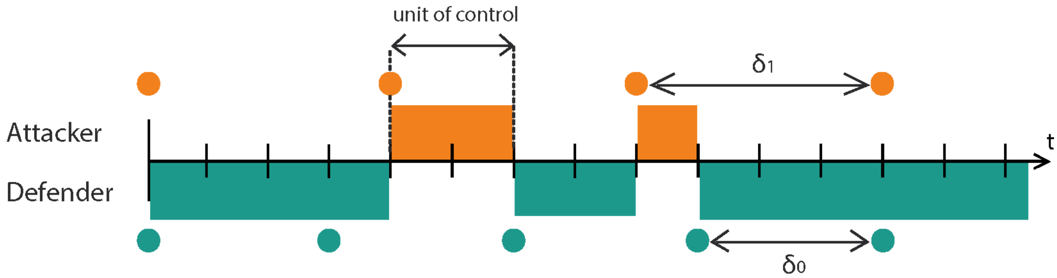
n: n is the n'th interval of the attacker, starting from interval 1.

ΔA : This is a function that denotes the length of a unit of control of the attacker in the n'th interval of the attacker.

lcm(a,b): The *lcm* of a and b is the least common multiple of a and b.

gcd(a,b): The *gcd* of a and b is the greatest common divider of a and b.

FIGUUR 8.2: Defining unit of control



We start by computing the gain of the attacker in a periodic game without phases. After that we introduce the phases. Next we compute the benefit of the attacker in function of the gain. Finally we adapt our gain and benefit calculation to include virus propagation.

hier staat al benefit dat nog niet is uitgelegd

We mag gebruikt worden in een wetenschappelijke tekst zolang de focus blijft op het werk en niet op de schrijver, FlipIt schrijvers gebruiken ook veel de we vorm

Computing the gain for an attacker of a normal FlipIt game

Case 1:
For $\delta_1 > \delta_0$ (The defender moves faster than the attacker.)

We consider a game in which both of the players start with a phase T_0 and phase T_1 equal to zero. Both players start their first move at $t = 0$. As previously stated (in the formal definition of the game and the introduction of different notations used throughout the paper), the defender has control in the beginning of the game at $t = 0$. If the two players move at the same time during the game, the moves cancel and no change of state happens.

eerder gedefinieerd dat elk spel begint op t=0

To compute the gain formula for the attacker, we need to calculate the amount of time that the attacker has control over the game from the start of the game up to time t. This can be done by computing all the units of control of the attacker up to time t and summing them.
The formula that we are going to compute will not be in function of the time but in function of the intervals of the attacker. By doing this we always have a whole

formule is niet in staat om dat te doen als functie van t unit of control. If time is used the last unit of control can be shortened. Because the game goes on indefinitely and because it is easier to compute a formula in function of the intervals of the attacker, the gain formula will be in function of the intervals of the attacker.

kan misschien beter uitgelegd worden

For that reason we divide our time line of our FlipIt game into different intervals of size δ_1. Therefore every time the attacker moves we have the start of a new interval. Considering that the defender will move faster than the attacker, he or she will at least move one time during the interval of the attacker. Because the attacker only moves at the start of his or her interval we can say that the defender will always end as being in control of the resource. . To calculate how long the unit of control of the attacker is in every interval, we only need to know how long the defender had control during that interval and substract it from the length of the interval of the attacker. This brings us to the next forumula to calculate the length of a unit of control in the n'th interval of the attacker. For every real δ_1 and $\delta_0 \in \mathbb{R}$ and every n $\in \mathbb{N}$ (including 0 in the set of natural numbers) :

tekening met voorbeeld

blabla en dan hebben we de volgende formule

definieer dat delta groter als nul is en niet negatief

$$\Delta A = [(1-n)\cdot\delta_1] mod \delta_0 \tag{8.1}$$

where n is the number of the n'th interval of the attacker where the length of the unit of control of the attacker is calculated.

The 1 - n is when we count beginning from 1. If we start counting starting at 0 we leave the 1 and the formula becomes:

$$\Delta A = [(-n)\cdot\delta_1] mod \delta_0 \tag{8.2}$$

For phases ..

beter beschrijven

Case 2: Alles omdraaien

Computing the average benefit for a normal FlipIt game

The average benefit of a player i is denoted by $\beta_i(t) = \gamma_i(t) - k_i/\delta_i$, which is equal to the fraction of time that the resource has been owned by player i, minus the cost rate for moving. For now we consider the cost rate equal to 0. $\gamma_i(t) = G_i(t)/t$ is the average gain rate of player i which is defined as the fraction of time that player i has control over the resource up to time t. As stated before the formula that will be computed is in function of δ_1. So parameter t will be in this case a multiple of the period of the attacker, hence a multiple of δ_1. In the rest of the paper the average benefit will be addressed as the benefit of the game for player i.

To be able to calculate the benefit of the attacker, we need to calculate the average gain rate of the attacker. We will look at this for two cases. One case where δ_0 and $\delta_1 \in \mathbb{Q}$ and the other one where δ_0 and $\delta_1 \in \mathbb{I}$.

Rational numbers ($\mathbb{Q}$): For δ_0 and $\delta_1 \in \mathbb{Q}$, a repetition of the game will occur. As every rational number is any number that can be expressed as the fraction p/q with p and q $\in \mathbb{Z}$ (integers), with the denominator q not equal to zero, it is possible to find the *lcm* of δ_0 and δ_1. The *lcm* is defined for all rational numbers as: $lcm(\frac{a}{b}, \frac{c}{d}) = \frac{a \cdot c}{gcd(b,d)} with$ []. When t is equal to the *lcm* of δ_0 and δ_1, both referentie players will move again at the same time and this can be mapped to the beginning of the game. Because we stated that at the end of the interval of the attacker, the defender is in control and because if two players move at the same time the moves are cancelled, we can map this to the beginning of the game. Since the game goes on infinite, to calculate the average gain of the attacker, it is sufficient to calculate the average gain of the attacker only during this repetition equal to the *lcm* of δ_0 and δ_1. So to calculate the gain of δ_0 and $\delta_1 \in \mathbb{Q}$ we need to calculate the amount of control units of the attacker that go into the length of time units equal to the *lcm* of δ_0 and δ_1. This will be equal to the amount of time that the period of the attacker goes into the *lcm*. We denote this by parameter p and define it as follow:

$$p = \frac{lcm(\delta_0, \delta_1)}{\delta_1} \tag{8.3}$$

After this calculation we summarize our units of control in function of p and divide it by the *lcm* of δ_0 and δ_1, which is the total amount of time for one cycle. This gives us the following formula of the benefit of the attacker with a cost rate equal to zero:

$$\beta_1 = \frac{\sum_{i=0}^{p} \{[(1-i) \cdot \delta_1] mod \delta_0\}}{lcm(\delta_0, \delta_1)} \tag{8.4}$$

Irrational numbers ($\mathbb{I}$): If δ_0 and/or $\delta_1 \in \mathbb{I}$: An irrational number $i \neq \frac{a}{b}$ with $b \in \mathbb{Z}$, a $\in \mathbb{N}$. Because we cannot write i as a fraction, this means that we cannot compute a common multiplier of δ_0 and δ_1. If their is no common multiplier the attacker and the defender won't move at one point on the same time, meaning that this does not result in a cycle. If we would have a cycle that means that there exists a number x that can be divided by δ_0 and δ_1. At $t = x$ both of the players will move at the same time, which is not possible because then their would be a cycle. Since their is no cycle it also means that no unit of control will be repeated two times. Every unit of control will have a distinct length. If it does that means that their is repetition, meaning again that their is a cycle. We can conclude that if we nog verder uitleggen have no cycle and no number will be repeated twice, that it will enumerate every number between 0 and the biggest interval (which is δ_0). *The reals are uncountable; that is: while both the set of all natural numbers and the set of all real numbers are infinite sets, there can be no one-to-one function from the real numbers to the natural numbers* [WikiPedia: real numbers] If they are uncountable that means that we cannot calculate the sum of all the numbers between 0 and the biggest interval. This is proved by the Cantor diagonalisation argument. Uncountable does not mean that we cannot order it. The Field of the real numbers is ordered.

What we can do is take the limit, count as many control units of time of the attacker and divide it by the greatest amount of time. We can see that this eventually will result to the solution given by the writers of FlipIt. [r/2]. Example delta1 Pi and delta0 1. Grafiek voor maken.

8.1.3 Formula with a virus propagation

Now we can define how we can use the previous formula to calculate the benefit of the attacker with a virus propagation. As mentioned before parameter d defines the virus propagation. It will take an amount of time d before the attacker gains control over all the resources. In the previous section we defined a formula to calculate each unit of control of the attacker. If the virus propagation takes d time before every resource is infected then this d has to be subtracted from each unit of control. (see figure 9.4). It may happen that the unit of control is less than d. The result of the substration will be a negative number in time. In this case this means that the defender has flipped all the resources before the attacker could gain control over all the resources. To calculate the benefit only the units of control bigger than 0 have to be summarized. So the formula becomes:

$$Gain_1 = \frac{\sum_{i=0}^{\delta_0}\{[(1-i)\cdot\delta_1]mod\delta_0 - d\} > 0\}}{delta_0 \cdot delta_1} \tag{8.5}$$

Hoofdstuk 9

APT

9.1 Advanced Persistent Threats

A targeted attack follows most of the time a serie of stages to attack its victim. This pattern of stages is also know as the Kill Chain, first mentioned by .. []. An APT will not always follow exact each step of this chain but it will give a good guideline of how an APT works.

1. **Reconnaissance**: During the first step of the Kill Chain an attacker will look for information to find an interesting victim. This information can be emailaddresses, IP addresses, conference information, anything that is available about the victim.

2. **Weaponization**: In the second stage the attacker will use an exploit and add a malicious playload to be send to the victim.

3. **Delivery**: The attacker will deliver his malicious code to the victim through different kins of intrusion methods. This can include email, usb stick, cd's, web, applications or other means.

4. **Exploitation**:The attacker executes the exploit, which is only relevant if the attacker uses an exploit.

5. **Installation**: The malware will be installed on the asset. This is only relevant if the attacker uses malware as a part of the attack.

6. **Command and Control**: The attacker will set up a command and control channel for remote manipulation of the victim.

7. **Actions on Objectives**: With "hands on keyboard" access, intruders accomplish their original goal.

nog uitbreiden, toevoegen that attackers will stay unnoticed for as long as possible or leave unnoticed with sensitive information


ATP
Cyber Kill Chain
1.
Reconnaissance
Harvesting email adresses, conference information, etc
2.
Weaponization
Coupling exploit with backdoor into deliverable payload
3.
Delivery
Delivering weaponized bundle to the victim via email, web, USB, etc
4.
Exploitation
Exploiting a vulnerability to execute code on victim's system
5.
Installation
Installing malware on the asset
6.
Command & Control
Command channel for remote manipulation of victim
7.
Actions on Objectives
With 'Hands on Keyboard' access, intruders accomplish their original goal

Hoofdstuk 10

FlipIt with virus propagation

10.1 FlipIt vs FlipIt with virus propagation

This chapter explains how to model a FlipIt game with a virus that propagates and infects the nodes in a network. First, this section explains the difference between FlipIt with and without a virus. Then, section 9.2 derives the formula to calculate the gain for a FlipIt game without a virus. After that section 9.3 introduces a modification to this formula to achieve an adapted gain formula for a FlipIt game with virus propagation. This allows us to derive the benefit formula.

In chapter 2 the FlipIt game was explained. This chapter starts from the specific case of a non-adaptive continuous FlipIt game where both players play a periodic strategy with a random phase. This choice is motivated by the assumption that in the practical situation of most organisations, the defence strategy is to periodically defend the network. This corresponds to a periodic defender strategy. To simplify the analysis in a first time, a periodic attacker strategy is assumed as well. Further research can investigate the effect of relaxing this assumption. check reff

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 time possible. After dropping a virus on the first node, it takes a while for the virus to infect the entire network. However, 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 have gained the control over the resource only when all the nodes of the network have been infected, i.e. the entire resource has been flipped.

10.2 Gain formula for a FlipIt game without virus propagation

In their FlipIt paper [], Marten van Dijk et al. , give a definition of the gain of a player i . This definition is however, in the best of our knowledge, not easy to adapt to the situation with virus propagation. Therefore, this section presents an alternative formula that defines a game by quantifying the amount of time each player has control.

The following notations will be used throughout the formal definition (see figure 9.1 for a graphic representation of some of the notations):

δ_D: This is the period of the defender. This denotes the length of the interval between two consecutive moves of the defender ($\delta_D > 0$).

δ_A: This is the period of the attacker. This denotes the length of the interval between two consecutive moves of the attacker ($\delta_A > 0$).

T_D: This denotes the phase of the defender that was chosen randomly and uniformly over the interval [0,δ_D].

T_A: This denotes the phase of the attacker that was chosen randomly and uniformly over the interval [0,δ_A].

Unit of control: Defined as the period between gaining (full) control and losing control over the resource.

n_D: The n'th interval of the defender, starting from interval 0.

n_A: The n'th interval of the attacker, starting from interval 0.

$\Delta Unit_D(n)$: This is a function that denotes the length of a unit of control of the defender in the n'th interval of the attacker or the defender depending on who is playing faster.

$\Delta Unit_A(n)$: This is a function that denotes the length of a unit of control of the attacker in the n'th interval of the attacker or the defender depending on who is playing faster.

lcm(a,b): The least common multiple of a and b.

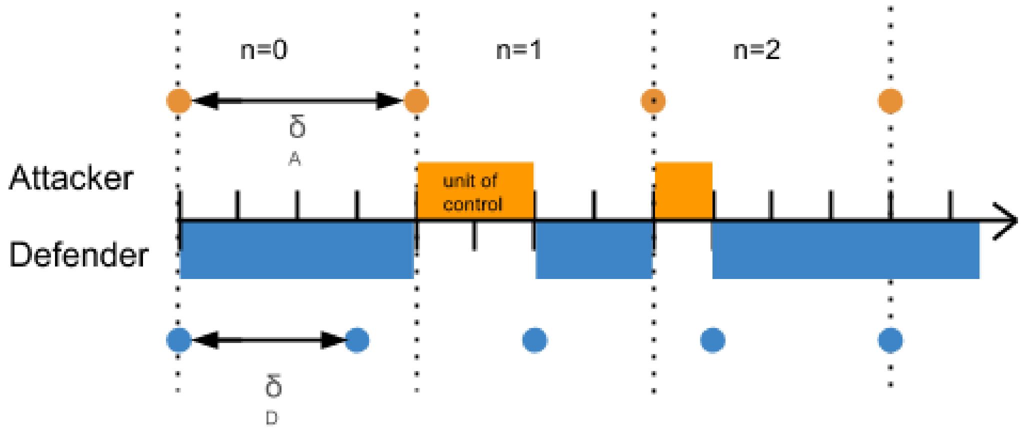
gcd(a,b): The greatest common divider of a and b.

$G_i(t)$: Gain of player i at time t. The gain of a player is defined as the total amount of time that a player has owned the resource since the start of the game up to time t. In the context of a FlipIt game with virus propagation, the whole network is seen as one resource. This resource is owned by the attacker if he has control over the entire network.

Average Gain: $\gamma_i(t) = G_i(t)/t$ is the average gain rate of player i which is defined as the fraction of time that player i has control over the resource up to time t.

Benefit: The average benefit of a player i is denoted by $\beta_i(t) = \gamma_i(t) - k_i/\delta_i$, which is equal to the fraction of time that the resource has been owned by player i, minus the cost rate for moving. For now we consider the cost rate equal to 0. In the rest of the paper the 'benefit' of the game for player i will be used as shorthand for 'average benefit'.

FIGUUR 10.1: Graphic representation of some of the notations



To compute the gain of a player in a periodic game without phases, two cases are considered: case 1 where the defender moves at least as fast as the attacker and case 2 where the attacker plays at least as fast as the defender. Next, the formula is enriched by introducing the phases.

Computing the gain for an attacker of a normal FlipIt game

Consider a game without phases, so in which both players start with a phase T_D and phase T_A equal to zero. Both players start their first move at $t = 0$. As previously stated (in the formal definition of the game and the introduction of different notations used throughout the paper), the defender has control in the beginning of the game at $t = 0$. For the remainder of the game, if the two players move at the same time during the game, the moves cancel each other out and no change of state happens.

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

To compute the gain formula for the attacker,the amount of time that the attacker has control over the resource from the start of the game up to time t has to be calculated. This can be done by computing the sum of all the units of control of the attacker up to time t.

To calculate a single unit of control of the attacker, the time line of the FlipIt game is divided into intervals of size δ_A. Every time the attacker moves we have the start of a new interval, with the attacker being in control, unless there is a simultaneous move with the defender. Considering that the defender moves at least as fast as the attacker, he or she will at least move one time during the interval of the attacker. Because the attacker only moves at the start of his or her interval we can say that the defender will always end as being in control of the resource at the end of an attacker's interval. .

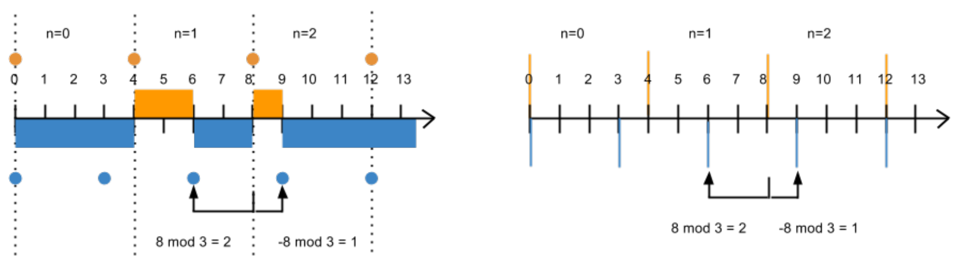
tekening met voorbeeld

To calculate how long the unit of control of the attacker is in the n'th interval, we only need to know how long the attacker has control over the resource before the defender moves in that interval. The start time of the n'th interval will be a multiple of the period of the attacker. Once the attacker has played, the time he can stay in control until the next move of the defender, depends on the time elapsed since the last time the defender played in the previous interval (the (n-1)th interval).

The time at which the defender plays in an interval is a multiple of its period. Once the defender has played for the last time in the (n-1)'th interval, the remaining time until the attacker will play can be calculated as $n \cdot \delta_a$ *modulo* δ_D which is equal to the remainder of $n \cdot \delta_A$ divided by δ_D. [referentie naar matworks: *http : //nl.mathworks.com/help/matlab/ref/mod.html*]The time the attacker will stay in control is then $\delta_D - n \cdot \delta_A$ *modulo* δ_D, which can also be calculated as $(-n \cdot \delta_A)$ *modulo* δ_D.

Figure 9.2 illustrates this graphically, for $\delta_A = 4$, $\delta_D = 3$ and the $n = 2$ interval. We see that in interval 1, the defender will stay 8 *modulo* $3 = 2$ in control, and so, in interval 2, the attacker will stay $1 = 3 - 2 = -(2 * 4)$ *modulo* 3 in control.

Figuur 10.2: Taking the modulo of a negative number



This brings us to the next formula to calculate the length of a unit of control in the n'th interval of the attacker.

For every positive and non zero real δ_A and $\delta_D \in \mathbb{R}$ and every $n \in \mathbb{N}$ (including 0 in the set of natural numbers) :

$$\Delta Unit_A(n_A) = [(-n_A) \cdot \delta_A] mod \delta_D \tag{10.1}$$

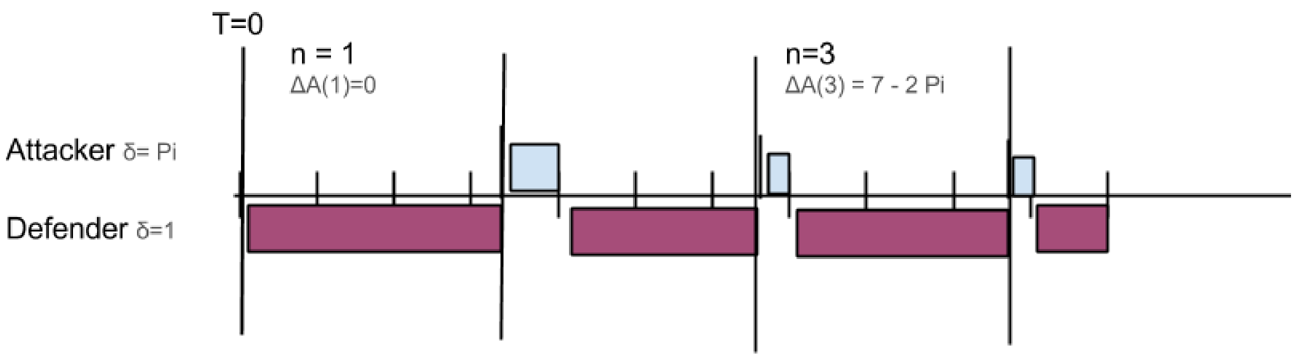
where n_A is the number of the n'th interval of the attacker starting from interval 0 where the length of the unit of control of the attacker is calculated.

The length of a unit of control of the defender is the remainder of the interval after the attacker loses control over the resource when the defender plays. This can be defines as follows for the n'th interval of the attacker:

$$\Delta Unit_D(n_A) = \delta_A - [(-n_A) \cdot \delta_A] mod \delta_D \tag{10.2}$$

An example: Figure 9.2 shows a FlipIt game were the period of the attacker is π and the period of the defender is 1. ... On figure 9.3 aanvullen

Figuur 10.3: Example for calculating the control unit in interval 3 for a FlipIt game with period defender = 1 and period attacker = pi



The gain formula can be calculated by taking all the units of control of the player up to an amount of p intervals of the attacker. The gain formula for the attacker is stated as follows:

$$Gain_A = \sum_{i=0}^{p} \{[(-i) \cdot \delta_A] mod \delta_D\} \tag{10.3}$$

where p is the number of units of control that have to be summed.

The gain of the defender is the sum of the units of control of the defender up to the same amount of p intervals of the attacker:

$$Gain_D = \sum_{i=0}^{p} \{\delta_A - [(-i) \cdot \delta_A] mod \delta_D\} \tag{10.4}$$

$$Gain_D = \sum_{i=0}^{p}\{\delta_A \cdot i\} - \sum_{i=0}^{p}\{[(-i) \cdot \delta_A] mod \delta_D\}\} \tag{10.5}$$

$$Gain_D = \delta_A \cdot p - \sum_{i=0}^{p}\{[(-i) \cdot \delta_A] mod \delta_D\}\} \tag{10.6}$$

Note: The gain formula is not in function of time t but the amount of p intervals of the attacker. This approach is chosen because it will result in whole units of control. It is possible to make a gain formula using the time, but this will result in a much more complicated function.

For phases ..

Case 2:
$\delta_D \geq \delta_A$ (The attacker moves at least as fast as the defender.)

For this case we use the same approach as in case 1 but with a small difference. To compute the unit of control of both players we divide the time line of the FlipIt game into intervals of size δ_D. The defender moves at the start of each interval, the end of the interval is the beginning of the next interval. Considering that the attacker will move at least as fast as the defender, he or she will move at least one time during the interval of the defender. Because the defender only moves in the beginning of each interval, the attacker will end as being in control of the resource.

If the unit of control of the defender need to be calculated, we only need to know how long it takes for the attacker to move in the interval. This can be done in the same way as in case 1 by taking the modulo of the negative of the beginning of the interval. The big difference with case 1 is when the length of the unit of control in the 0'th interval is calculated. Because the defender always has control in the beginning of the game, the first interval is computed in a different way. The unit of control of the defender in the 0'th interval is equal to the length of the period of the attacker, since from that moment the attacker takes control.

For every positive and non zero real δ_A and $\delta_D \in \mathbb{R}$ and every n $\in \mathbb{N}$ (including 0 in the set of natural numbers) :

for $n_A = 0$

$$\Delta Unit_D(n_D) = \delta_A \tag{10.7}$$

for $n_A > 0$

$$\Delta Unit_D(n_D) = [(-n_D) \cdot \delta_D] mod \delta_A \tag{10.8}$$

where n_D is the number of the n'th interval of the defender starting form interval 0 where the length of the unit of control of the defender is calculated.

The length of a unit of control of the attacker is the remainder of the interval after the defender loses control over the resource when the attacker plays. This can be defined as follows for the n'th interval of the defender:

$$\Delta Unit_A(n_D) = \delta_D - \Delta Unit_D(n_D) \tag{10.9}$$

The gain formula for $\delta_D \geq \delta_A$ is calculated in the same manner as case 1:

The gain of the defender is the sum of the units of control of the defender up to the same amount of p intervals of the attacker:

$$Gain_D = \sum_{i=1}^{p} \{[(-i) \cdot \delta_D] mod \delta_A\} \quad with \quad i = 0 \quad Gain_D = \delta_A \tag{10.10}$$

and for the attacker:

$$Gain_A = \delta_D \cdot p - \sum_{i=1}^{p} \{[(-i) \cdot \delta_D] mod \delta_A\} \quad with \quad i = 0 \quad Gain_A = \delta_D - \delta_A \tag{10.11}$$

10.3 Gain and benefit formula for a FlipIt game with virus propagation

The formulas from the previous section can now be adapted to calculate the gain of both players in a FlipIt game with virus propagation. As mentioned before, the attacker will try to infect all the nodes in the network. He will do this by flipping the node in the graph that can infect all the nodes in the shortest time possible. After dropping a virus on the first node, it takes a while for the virus to infect the entire network. The time that it takes for the virus to infect every node will be denoted as parameter d. 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. This can be measured by a method explained in section [matrix berekeningen]. Assume that an attacker attacks at time t, then only at time $t+d$ he gains control over the entire network. 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 control over the entire network. Using this parameter d, a FlipIt game with virus propagation can be modelled.

The previous section defined a formula to calculate each unit of control of the attacker and the defender for two cases. If the virus propagation takes d time before every resource is infected then this d has to be subtracted from each unit of control. (see figure 7.4 9.4). It may happen that the unit of control is less than d. In that case, the result of the substration will be a negative number, meaning that the defender has flipped all the resources before the attacker could gain control over all the resources. To calculate the gain only the units of control bigger than 0 have to be summed. So the formula becomes:

For $\delta_A \geq \delta_D$:

$$Gain_A = \sum_{i=0}^{p} \{[(-i) \cdot \delta_A] mod \delta_D - d\} > 0\} \tag{10.12}$$

where p is the number of units of control that have to be summed.

The gain of the defender is equal to the amount of time that the attacker is not in control of the resource. So the formula for the defender becomes:

$$Gain_D = p \cdot \delta_A - \sum_{i=0}^{p} \{[(-i) \cdot \delta_A] mod \delta_D - d\} > 0\} \tag{10.13}$$

For $\delta_D \geq \delta_A$:

$$Gain_A = \delta_D \cdot p - \sum_{i=1}^{p} \{[(-i) \cdot \delta_D] mod \delta_A - d\} > 0\} \tag{10.14a}$$

$$with \;\; i = 0 \;\; Gain_A = \delta_D - (\delta_A - d) > 0 \tag{10.14b}$$

(10.14c)

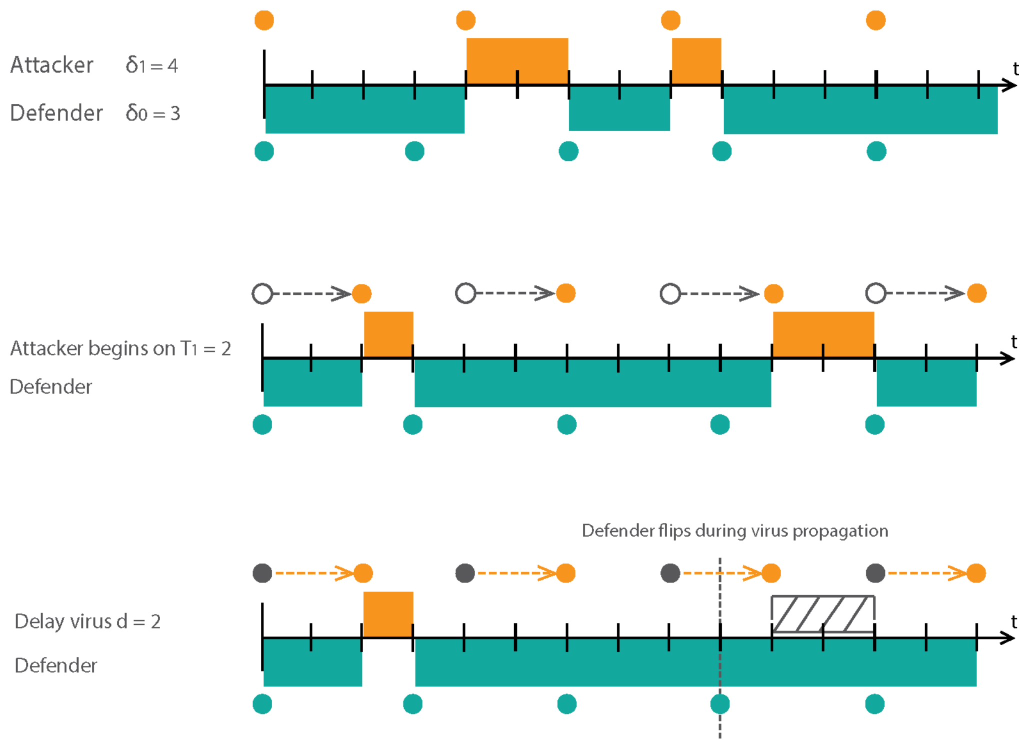
and for the defender:

$$Gain_D = \sum_{i=1}^{p} \{[(-i) \cdot \delta_D] mod \delta_A - d\} > 0\} \tag{10.15a}$$

$$with \;\; i = 0 \;\; Gain_D = \delta_A - d > 0 \tag{10.15b}$$

(10.15c)

Figuur 10.4: Difference in a FlipIt game between delay caused by a virus and a phase bigger than zero for the Attacker



Computing the benefit of a FlipIt game with virus propagation

Calculating the benefit of both players, requires calculating the average gain rate of both players. To compute the benefit the value of parameter p needs to be determined. Two cases can be considered: one case where δ_D and $\delta_A \in \mathbb{Q}$ and the other one where δ_D and $\delta_A \in \mathbb{I}$. In both cases we first calculate the benefit of the attacker in case that the defender moves at least as fast as the attacker. The benefit of the defender will be BenD = 1 - BenA. The benefit of both players for the case where the defender moves as least as fast as the defender is done in a similar way.

Rational numbers ($\mathbb{Q}$): When δ_D and δ_A are rational numbers, after a number of intervals (namely their least common multiple), the same pattern of intervals will be repeated over and over again. Why? A rational number is a number that can be expressed as the fraction p/q with p and q $\in \mathbb{Z}$ (integers), with the denominator q not equal to zero, it is possible to find the *lcm* of δ_D and δ_A. The *lcm* is defined for all rational numbers as: $lcm(\frac{a}{b}, \frac{c}{d}) = \frac{lcm(a,c)}{gcd(b,d)} with$ []. When t is equal to the *lcm* of δ_D and δ_A, both players will move again at the same time and this can be mapped to the beginning of the game. Because we stated that at the end of the interval of the attacker, the defender is in control and because if two players move at the same time the moves cancel each other out, we can map this to the beginning of the game. Since the game goes on infinitely, to calculate the average gain of the attacker, it is sufficient to calculate the average gain of the attacker only during a period of time equal to the *lcm* of δ_D and δ_A. Since *lcm* is a multiple of δ_D and δ_A, there is a number p so that *lcm* =p $\cdot\delta_A$, meaning that the attacker will have played p times. p can be defined as follows:

referentie

$$p = \frac{lcm(\delta_D, \delta_A)}{\delta_A} \tag{10.16}$$

This results in the following formula for the benefit of the attacker with a cost rate equal to zero:

$$\beta_A = \frac{\sum_{i=0}^{p} [[(-i) \cdot \delta_A] mod \delta_D - d] > 0}{lcm(\delta_D, \delta_A)} \tag{10.17}$$

As stated before [], the benefit of the attacker and the benefit of the defender add up to 1 ($\beta_A + \beta_B = 1$). The benefit of the defender can be writen as follows:

$$\beta_D = 1 - \beta_A \tag{10.18}$$

Irrational numbers ($\mathbb{I}$): If δ_D and/or $\delta_A \in \mathbb{I}$: An irrational number $i \neq \frac{a}{b}$ with $b \in \mathbb{Z}$, a $\in \mathbb{N}$.

Two cases can be distinguished. (A) $\frac{\delta_D}{\delta_A}$ is a rational number a/b with a <= b. In that case, after b intervals, the pattern will repeat itself.

(B)If either δ_D and δ_A cannot be written as a fraction, and they are no multiple of each other, the least common multiplier cannot be calculated. Moreover, there

will be no repeating pattern. If both players move at one point in the game at the same moment, this point of time has to be a multiple of the period of the attacker and a multiple of the period of the defender. But because there is no least common multiple, no such point of time exists during the game. If both players never play at the same moment, it is not possible to have a repeated pattern because no mapping to the beginning of the game can occur. Additionly two unit of controls with the same length cannot exist. This would mean that the game has a repeated pattern, which is not possible.
The game will go on forever, if no repeating pattern occurs and it would keep on generating units of control with different lengths. This implies that if the game goes on forever, every length between 0 and the smallest interval (which is δ_D) will be generated. To calculate the benefit we want to summarize the unit of controls up to a number of interval p. Considering that the game goes on forever without repetition we cannot rely on the fact that the benefit can also be calculated only during the repetition. Calculating the benefit of a game without repetition would imply that all the unit of control to infinite have to be calculated. This implicates that all the numbers between 0 and δ_D have to be summed but this is impossible. *The reals are uncountable; that is: while both the set of all natural numbers and the set of all real numbers are infinite sets, there can be no one-to-one function from the real numbers to the natural numbers* [WikiPedia: real numbers] If they are uncountable that means that we cannot calculate the sum of all the numbers between 0 and the biggest interval. This is proved by the Cantor diagonalisation argument. Uncountable does not mean that we cannot order it. The Field of the real numbers is ordered.

nog mooie tekst schrijven in engels: hier essentie

We kunnen de benefit wel benaderen door een zo groot mogelijke som te nemen van de unit of controls. Uit deze benadering is af te leiden waar de verhouding naartoe zou gaan als de limiet zou genomen worden. -> laten zien met een voorbeeld van Pi en 1.

Hoofdstuk 11

Formula

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.

11.1 Explaining difference between FlipIt with and without virus propagation

zelfde als in vorige chapter:

In chapter 2 the FlipIt game was explained. This chapter starts from the specific case of a non-adaptive continuous FlipIt game where both players play a periodic strategy with a random phase. This choice is motivated by the assumption that in the practical situation of most organisations, the defence strategy is to periodically defend the network. This corresponds to a periodic defender strategy. To simplify the analysis in a first time, a periodic attacker strategy is assumed as well. Further research can investigate the effect of relaxing this assumption.

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 time possible. After dropping a virus on the first node, it takes a while for the virus to infect the entire network. However, 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 have gained the control over the resource only when all the nodes of the network have been infected, i.e. the entire resource has been flipped.

After dropping a virus on the first resource, it takes a while for the virus to infect the entire network. The time that it takes for the virus to infect every node will be denoted as parameter d. 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. This can be measured by a method that we will explain in section []. Assume that an attacker attacks at time t, then only at time t + d he gains control over the entire network. 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 control over the entire network. Using this parameter d, a FlipIt game with virus propagation can be modelled.

11.2 Benefit for FlipIt game with virus propagation

Periodic Game with delay for the attacker:

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.

However, because of the delay, the maximal time of control is reduced to $\delta_D - d$. There is a probability of r that the attacker will move in the interval of the defender. 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. There is $\dfrac{\delta_D - d}{\delta_D}$ probability that the attacker will move soon enough which gives the attacker a 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 average gain rate of the attacker can be expressed as follows if we look at one interval of the defender:

$$\beta_A(\alpha_D, \alpha_A) = \frac{1}{\delta_D}[\frac{\delta_D}{\delta_A} \cdot \frac{\delta_D - d}{\delta_D} \cdot \frac{\delta_D - d}{2} + \frac{\delta_D}{\delta_A} \cdot \frac{d}{\delta_D} \cdot 0] \tag{11.1}$$

To complete the formula to derive the benefit function, the cost of moving is added. In the second formula we can see the formula of the original FlipIt game.

$$\beta_A(\alpha_D, \alpha_A) = \frac{(\delta_D - d)^2}{2 \cdot \delta_D \delta_A} - k_A \alpha_A \tag{11.2}$$

$$\beta_A(\alpha_D, \alpha_A) = \frac{\delta_D}{2 \cdot \delta_A} - k_A\alpha_A + \frac{d}{\delta_A} + \frac{d^2}{2 \cdot \delta_A\delta_D} \tag{11.3}$$

The benefit of the defender is expressed as follows:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{(\delta_D - d)^2}{2 \cdot \delta_D\delta_A} - k_D\alpha_D \tag{11.4}$$

11.3 something

Periodic Game with delay for the attacker:

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 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.

However, because of the delay, the maximal time of control is reduced to $\delta_D - d$. If we consider a duration of $\delta_D \cdot \delta_A$ the attacker will play δ_D times. If the attacker plays soon enough it will get a gain of $\frac{\delta_D - d}{2}$ in $\delta_D - d$ of the cases. In d cases it will receive a gain of zero. This is the case were the duration of the delay causes the defender to play before the attacker can get control over the resource. So the gain of the attacker can be expressed as follows:

$$Gain = \frac{\delta_D - d}{2} \cdot (\delta_D - d) + 0 \cdot d = \frac{\delta_D - d}{2} \cdot (\delta_D - d) \tag{11.5}$$

The benefit of the attacker can be expressed as follows

$$\beta_A(\alpha_D, \alpha_A) = \frac{(\delta_D - d)^2}{2 \cdot \delta_D\delta_A} + k_A \cdot \alpha_A \tag{11.6}$$

$$\beta_A(\alpha_D, \alpha_A) = \frac{\delta_D}{2 \cdot \delta_A} + k_A \cdot \alpha_A + \frac{d}{\delta_A} + \frac{d^2}{2 \cdot \delta_A\delta_D} \tag{11.7}$$

The benefit of the defender is then:

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{(\delta_D - d)^2}{2 \cdot \delta_D \cdot \delta_A} + k_D \cdot \alpha_D \tag{11.8}$$

$$\beta_D(\alpha_D, \alpha_A) = 1 - \frac{\delta_D}{2 \cdot \delta_A} + k_D \cdot \alpha_D - \frac{d}{\delta_A} - \frac{d^2}{2 \cdot \delta_A\delta_D} \tag{11.9}$$

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

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 $\frac{\delta_D}{\delta_A} = (1/r)$. Given that the defender moves within the interval, he moves exactly once within the interval (since$\delta_A \leq \delta_D$) and his move is distributed uniformly at random.

If we consider a duration of $\delta_A \cdot \delta_D$ there is a probability of $\frac{\delta_A}{\delta_D}$ that the defender moves within the interval of the attacker. The defender will then receive an average gain of $\frac{\delta_A}{2}$. There is $1 - \frac{\delta_A}{\delta_D}$ probability that the defender will not move in the interval of the attacker and so the defender will receive no gain. The benefit can be expressed as follows when the defender plays δ_D times during a duration of $\delta_A \cdot \delta_D$:

$$\beta_D(\alpha_D, \alpha_A) = \frac{1}{\delta_A \delta_D} \cdot \delta_D \cdot [\frac{\delta_A}{\delta_D} \cdot \frac{\delta_A}{2} + [1 - \frac{\delta_A}{\delta_D}] \cdot 0] + k_D \cdot \alpha_D \tag{11.10}$$

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

same as the FlipIt solution

(11.11)

However, because of the delay, the maximal time of control of the defender is increased by d. In other words, the defender has some benefit time of d before the attacker really gains control over the resource, meaning that the attacker gains control only after $\delta_A + d$ instead of after δ_A. So, when the defender plays, with a probability of $\frac{\delta_A}{\delta_D}$, the expected gain of the defender's control in this interval would be more than half of the period δ_A: it is $\frac{\delta_A + d}{2}$. There is $1 - \frac{\delta_A}{\delta_D}$ probability that the defender will not move in the interval of the attacker but because of the delay the defender will receive a gain of d. So the benefit of the defender can be expressed as:

$$\beta_D(\alpha_D, \alpha_A) = \frac{1}{\delta_A \delta_D} \cdot \delta_D \cdot [\frac{\delta_A}{\delta_D} \cdot \frac{\delta_A + d}{2} + [1 - \frac{\delta_A}{\delta_D}] \cdot d] + k_D \cdot \alpha_D \tag{11.12}$$

$$\beta_D(\alpha_D, \alpha_A) = \frac{\delta_A - d}{2 \cdot \delta_D} + \frac{d}{\delta_A} + k_D \cdot \alpha_D \tag{11.13}$$

The benefit of the attacker is expressed as follows:

$$\beta_A(\alpha_D, \alpha_A) = 1 - [\frac{\delta_A - d}{2 \cdot \delta_D} + \frac{d}{\delta_A}] + k_A \cdot \alpha_A \tag{11.14}$$

Hoofdstuk 12

The Final Chapter

12.1 chap

Hoofdstuk 13

Conclusion

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

13.1 trala

Bijlagen

Bijlage A

The First Appendix

Appendices hold useful data which is not essential to understand the work done in the master thesis. An example is a (program) source. An appendix can also have sections as well as figures and references[**?**].

A.1 More Lorem

Bijlage B

The Last Appendix

Appendices are numbered with letters, but the sections and subsections use arabic numerals, as can be seen below.

B.1 Lorem 20-24

Fiche masterproef

Student: Sophie Marien

Titel: Gametheory and Cybersecurity: a study FlipIt and multiple resources

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