Modelling Ransomware Attacks using POMDPs

Henry R.K. Skeoch

University College London

Abstract

Ransomware attacks have become a major source of criminal revenue generation and a nuisance to a wide range of organisations. A particular challenge in planning and improving defences against these attacks is that key overall corporate decision makers may not possess sufficient technical computing knowledge to fully understand the attack chain. This work introduces a Partially Observable Markov Decision Process (POMDP) model of a ransomware attack on a network and explains how it can be used to assist non-technical decision makers in evaluating optimal defence strategies. It is also demonstrated how the model might help insurance providers rapidly model specific claim probabilities for individual risks.

I Introduction

Ransomware is a commonly used term for computer code that uses encryption to compromise the availability of files and/or a system with the aim of extracting a ransom from the victim. The concept of ransomware has been discussed in the literature for at least 25 years with Young and Yung(1996)[1] introducing the concept of 'cryptovirology' where they envisaged encryption being used offensively to extort money from a system owner. The literature on ransomware spans the fields of computer science, crime science and economics. Ransomware was largely of theoretical interest until relatively recently owing to the difficulty for criminal enterprises to extract payments from victims. Legitimate financial institutions are in general prohibited from engaging in or facilitating criminal transactions and this rendered cross-border extraction of ransom payments problematic. However, the development of crypto-currencies has facilitated pseudonymous monetary transactions outside conventional financial channels. This has proved to be an effective enabler of cyber-dependent[1] crime such as ransomware.

A Trend Micro white paper in conjunction with Osterman Research[3] reported that 50% of surveyed firms lacked the capability to prevent or detect ransomware attacks. The increased use of cyber-insurance by firms risks transferring the burden of costs from the attacked business to the insurer depending on the exact terms of insurance coverage. The ability to model the extent to which an individual risk[2] may be affected by ransomware is of critical importance to a responsible insurer's underwriting strategy in determining the expected frequency of claims and also the severity[3]. Use of economic models such as the one presented in this research may help insurers better understand how to price the risks of ransomware and thus provide better coverage for firms.

I.1 Modelling ransomware requires a clear, disciplined approach

The potential costs of a ransomware attack may be mapped using the popular confidentiality-integrity-availability (CIA) framework. Assuming the encryption process is perfectly reversible with the appropriate key, the integrity of the information may be unaffected, though there is always the risk of corruption. If the ransomware allows the attacker access to or exfiltration of information, then there is a potential for breach of confidentiality. The encryption methodology can vary in sophistication,

[1] See Wall(2005)[2] for an excellent taxonomy of cybercrime

[2] In insurance, it is common to refer to policyholders as risks

[3] Average loss per claim

but even a relatively rudimentary encryption methodology would prove hard to crack within a limited timeframe given to pay the ransom (unless the ransomware is a common variant using a key that has already been cracked). The interaction between the attacker and defender is potentially nuanced and complex in a ransomware attack relative to other cybercrimes. In ransomware, there may be direct interaction and bargaining between an attacker and defender, whereas other malware such as spyware, keyloggers and so on may compromise the confidentiality of information or its integrity, but the interaction between attacker and defender is usually indirect.

The motivation for the research depicted in this paper is to develop a model for ransomware infections and defence that has broad accessibility and applicability. It is important, however, to ground this against an established and well-studied class of models to allow for a range of existing techniques to be used to study the model output. This approach is inherently vulnerable to the criticism of abstraction from real world cases. However, a model that perfectly replicates every detail of a system is likely to be difficult to efficiently solve. Thus, there is a balance to be struck between choosing a set of sufficiently sparse parameters to depict the problem and losing practical significance.

A particular challenge in modelling ransomware is finding a framework to capture the intricate architecture of different networks and different attackers. Partially Observable Markov Decision Process models appear to be particularly useful for describing a ransomware infection as they can capture uncertainty about the state of a system from the perspective of the observer. Further the transition structure allows for either deterministic or stochastic outcomes or a mixture of both. Whilst the representation of a network system within such models is relatively simple compared with the complex structure of protocols, privileges and interaction that comprise a network, a POMDP model at least captures the core features of the system architecture in a way that most conventional game theory based models may not.

I.2 Distinguishing different ransomware attacks

For the purposes of this research, ransomware is considered to take two forms: wormlike malware without attacker interaction and malware launched by a strategic attacker. The former automated threat can be countered by antivirus companies updating detection signatures and software vendors patching known exploited vulnerabilities. This type of ransomware is economically similar to a mass marketing effort, where a criminal enterprise hopes to gain large numbers of small ransoms. This has been well researched and documented, with backup of data often used as the key defensive strategy. The wide availability of secure (in so far as anything can be) cloud storage mitigates to some extent against risks of the loss of information availability but does not solve the risk of a breach of confidentiality. For individuals, ransomware insurance may be of value in covering the costs of a replacement device should an expensive piece of equipment be rendered inoperable (especially if a backup of data is available).

Backup is however only partially effective for the latter type of ransomware, which targets enterprise networks. Typically, this type of ransomware is introduced via either a malicious email attachment; via direct unauthorised network access (Remote Desktop Protocol, for example); or by exploiting a vulnerability in a system. Under the assumption that the main objective of such an attack is to render key network nodes unavailable in the hope of extracting a ransom, if a prior backup exists it will be of a de facto vulnerable configuration that if restored may be immediately compromised again. There may be a mitigating patch or configuration alteration available, but this is not guaranteed. In the event that the attacked organisation places most weight on pure information, a backup is useful. However, this may not solve the potentially significant financial risk of a loss of business operations. The most significant public example of this is the shipping conglomerate, Maersk, who suffered a global logistics outage as a result of the NotPetya malware[4].

I.3 Ransomware may incur reparative costs

Once an organisation is aware it has been compromised, it may enlist the help of a specialist company providing 'post-breach services'. This may be paid for either by the victim itself, or increasingly

[4]See, for example, Greenberg (2018)[4] for an interesting account.

commonly by an insurer as part of a cyber-insurance policy. The trade-offs between the cost and benefits of these services is an emerging but potentially fruitful area of research. The decision process following a ransomware infection ranges from attempting only the minimum remedial actions needed to clear the immediate ransomware infection (including paying the demanded ransom) to a complete replacement of all information technology infrastructure including a 'clean install' of all operating systems and software. A rational firm paying for the clean-up itself would choose the minimum cost needed to contain its eventual potential loss. Losses include potential third party claims in case of data leakage, loss of turnover due to business interruption and direct expenses related to combatting the ransomware infection. An interesting question emerges when an insurer is paying for the cost of the clean-up. In this case, subject to the limits of policy and the risks of affecting future premia, there is a potential incentive to spend more than the minimum amount if the firm is not paying for the post-breach costs itself. It is clearly in the interests of the post-breach specialist to maximise its income from such an operation.

II Related work

II.1 Game theory models of ransomware infection

A small but high quality body of literature around the economics of ransomware has developed, chiefly organised around a game theoretic treatment of bulk ransomware attacks. Laszka et al (2017)[5] model ransomware as a multistage, multidefender game with mitigation via backup. In the game, the first stage is organisations and attackers choosing their backup and attack efforts respectively. In stage two, each organisation becomes compromised; those falling victim decide whether to pay the ransom. August et al (2017)[6] provide an extremely thorough economic treatment of the problem of software with vulnerabilities potentially exploitable to deliver ransomware. They examine a downstream endogenous recovery decision that influences an upstream security decision. They note that a limitation of prior literature is that the possibility of negative security externalities is not captured. The work is particularly focused around the trade-off between software pricing and potential for ransom, which while of theoretical interest is practically less intuitive as the monetary cost of software is just one factor governing its adoption or utilisation. Cartwright et al (2019)[7] develop two prior game theoretic models of kidnapping: Selten and Lapan & Sandler. Their set of payoffs comprises: criminal does not infect computer; release of files for ransom & not caught; files destroyed & not caught; criminal caught after release of files; criminal caught after destroying files. Li and Liao(2020)[8] consider a multi-stage game. In stage 1, the attacker launches ransomware attacks on N victims. In stage 2, after observing random, R, victims decide whether or not to pay it. In stages 3 & 4, the attacker follows up with decision making. An interesting innovation by Li and Liao is the introduction of a reputation score for the ransomware originator. Ryan et al (2021)[9] construct an asymmetric non-cooperative two player game to consider how the development of targeted ransomware has affected the dynamics of ransomware negotiations. Galinkin (2021)[10] frames the ransomware defence problem as a lottery and considers how best to remove the incentives based on data from actual ransomware attacks. Yin et al (2021)[11] conduct a game-theoretic analysis of ransomware via attacker-defender and defender-insurer games. They find that backup strategies are abandoned when recovery becomes too expensive and that the introduction of insurance leads to moral hazard.

II.2 POMDP models of penetration testing

There is a reasonably developed, though arguably fairly concentrated, body of literature on the use of partially observable Markov decision process models (POMDPs) for penetration testing. This work is organised around the identification of potential attack paths within a system from a defensive perspective. However, this methodology is equally applicable to the decisions of an attacker albeit the attacker may be more risk averse with regard to potential detection. One possible reason why this study is not more popular is that vulnerabilities in systems can be esoteric and the POMDP model therefore both too general and abstract to usefully model the cases. However, for a broad economic analysis of systems vulnerability, these models may yield useful insights. Sarraute et al (2012)[12] represents

an early use of POMDPs for modelling penetration testing by considering the planning of attacks under uncertainty. Sarraute et al (2013)[13] refines the authors' earlier paper, but concludes that in general penetration testing is not POMDP solving, for the reason that the specificity of the models is inherently limiting set against the continually evolving information security landscape. Hoffman (2015)[14] provides a taxonomy of models in respect of the previous research, but again highlights the limitations of decision models in fully capturing human behaviour. Mehta et al (2016)[15] discuss how POMDPs can be used to inform resilient systems design, which is clearly of relevance to understanding how to defend against ransomware attacks. Ghanem and Chen (2020)[16] highlight the value in using automated reinforcement learning to replicate and analysing complex penetration tests far faster than even an expert human might be able to. Schwartz et al (2020)[17] present two different POMDP-based penetration testing models, though the work appears relatively abstract but the introduction of a discount factor is interesting.

II.3 Post-breach services

An interesting consideration in cyber-insurance policies is coverage of incident response services. Woods and Böhme (2021)[18] conduct (to the best of our knowledge) the first survey of how insurers address this particular problem. They find that insurers tends to nominate a panel of firms to provide services to insured parties, split between legal, forensics and communications experts. The panel sizes range from just 5 firms (Allianz) to 50 (AIG) within the top 20 US cyber-insurance carriers who make such information public. Woods and Böhme highlight that the question as to whether insurers have resulted in a worsening of the 2021 ransomware epidemic is an empirical one to which they are not aware of any answers. Further, they fail to distil any stylised facts about ransom procedures finding "considerable variation across insurers and providers". Without such information, it is arguably difficult for firms to plan a strategy *ex ante* and it is this decision making process that our model aims to assist with.

II.4 Business continuity and recovery

Business continuity insurance is a long-standing line of insurance, which traditionally covered computer systems and data records under the 'all other contents' definition under 'property damage' as noted in Glynn et al (2020)[19]. Glynn et al (2020) also note that "commercial combined policies have generally sought to exclude hacking attacks and losses flowing from viruses, corruption of data, etc." This clearly therefore excludes ransomware attacks, which are arguably better covered under a cyber-insurance policy. Filiz et al (2021)[20] conduct an interesting study into the effectiveness of ransomware decryption tools; the malware in this study is largely of that encountered in the wild rather than the targeted strains covered by the research in this paper.

II.5 Network malware models

Jacob et al (2008)[21] present an interesting treatment of the issues that might need to be addressed in an automata model of malware, in particular interaction and concurrency. Dalla Preda and Di Giusto (2011)[22] offer a formalisation of this thinking via the κ-calculus. Cam (2017)[23] develops a combined POMDP/logistic regression model for minimising the impact of a malware infection. Liu (2019)[24] presents a thorough theoretic analysis of ransomware spreading across a network incorporating its specific topology using an adjacency matrix. The model assumes that the dynamic state of each network node is statistically dependent on the states of its neighbouring nodes. Hu et al (2020)[25] use Bayesian attack graphs to model the interactions between a multi-stage attacker and a network, formulating the defence problem as a POMDP.

II.6 Empirical cyber-insurance studies

A growing body of literature aims to answer the question as to how cyber-insurers behave in practice. Woods et al (2017)[26] provide a useful survey of cyber-insurance proposal forms, which addresses the questions that insurers ask those seeking cover. Romanosky et al (2019)[27] use publicly available

US carrier submissions to analyse how cyber-insurance is priced; from this work, it appears that the methodology used in industry is not particularly sophisticated and could be greatly improved. This is a significant modelling challenge that has seen notable growth in start-up firms attempting to capitalise on related insurer demand. Nurse et al (2020)[28] present a qualitative study of data and processes driving underwriting of cyber-insurance, based on extensive interviews with market participants.

III Model

III.1 Problem statement and economic considerations

We define a ransomware attack as the introduction of a malicious process that uses encryption to compromise the availability of a system by attacking the integrity of the system, manipulating existing processes and resources, potentially with loss of confidentiality as well. Confidentiality, integrity and availability are harder to represent mathematically than monetary costs. The economic concept of utility is helpful in this situation as it provides a way to describe the preferences of a decision maker (often called an agent in the economics literature). A so-called multi-attribute utility function can be defined to represent the preferences of the decision maker in this problem:

$$U_{defender} = U(\kappa, \iota, \alpha) \tag{1}$$

where κ represents confidentiality, ι integrity and α availability. $U(\kappa, \iota, \alpha)$ is a multi-attribute utility function, which may vary according to the preferences of the defender. We assume for simplicity that this takes the value of 1 for a system operating according to its specified parameters. This framework accounts for the expected benefits of security investment including insurance coverage in a rigorous manner.

The concept of integrity is important in the attack as for a large-scale ransomware attack to be effective, a process needs to be introduced into the target system with sufficient privileges to effect encryption of key files beyond the privileges of the initially compromised user. Targeted resources may include credentials (passwords, keys etc), configuration files (for access control or firewalls). Manipulation of firewalls is particularly important if the attacker seeks to exfiltrate data from the attacked system, though this is not likely to be the primary motivation of a ransomware attack but rather a strategy by the attacker to increase the likelihood of ransom payment. Figure 1 summarises the utility impact of various different types of attack.

Figure 1: Attack impact on utility

Attack	Confidentiality	Integrity	Availability
Data Breach	x	x	
Locker Ransomware		x	x
Double-extortion	x	x	x

III.2 POMDP Model Structure

A POMDP (see Kälbling et al (1998)[29]) is characterised as a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$, where S : Set of states, A : Actions, T : Conditional transition probabilities between states, $R : S \times A \to \mathbb{R}$ (reward function), Ω : Observations, O : Conditional observation probabilities, γ : Discount function
This research introduces a POMDP for an agent defending a network of machines against a ransomware attack. The model POMDP code structure is based on the Julia package *POMDPs.jl*[30]. Julia has several advantages for this type of work: legibility of code and outputs via using symbols to represent key parameters, speed of computation and finally the ability to define custom types.

III.2.1 Model structure

The set of states for the POMDP model are S = {: clean, : infected, : locked, : offline}[5]. The rationale for choosing these states is that when a machine is compromised, it is not certain that the files contained on it will immediately be encrypted or that it will be locked[6]. In a targeted ransomware attack, the attacker may wish to compromise multiple systems within the network before attempting to extort a ransom. The model states may apply to the system as a whole in the case of a single machine, or to individual machines in the multi-machine cases contained within a vector. The 'offline' state is a terminal state for the single machine case and if a systemically critical machine such as a domain controller is offline in the multi-machine case.

The overall set of available actions is defined as A = {: observe, : repair, : shutdown, : pay}. Within each state, only certain actions are available (Figure 2). The actions apply to the system as a whole rather than individual machines. This sacrifices some potential realism but has the benefit of significantly reducing potential dimensional complexity in the model transition structure. For the purposes of this work, the actions {: shutdown, : pay} are assumed to be terminal. Thus, paying the ransom restores the system to its original clean state without possible reinfection. The monetary reward struc-

Figure 2: Model actions

State	Observe	Repair	Shutdown	Pay
Clean	x	(x)		
Infected	x	x		
Locked			x	x
Offline				

ture for the model is depicted in Figure 3. In addition to monetary rewards, the reward function can also update the utility function, U_{defender} based on the action taken and the resultant state, s'. The set

Figure 3: Model reward structure

Parameter	Description
$r_{observe}$	Cost of observation
r^{+}_{repair}	Cost of successful repair
r^{-}_{repair}	Cost of unsuccessful repair
$r_{shutdown}$	Cost of shutting down system
r_{ransom}	Cost of ransom payment

of observations, $\Omega \in S$, are equivalent to the model states. These are accompanied by an observation accuracy parameter, $p_{obs} = [0 \rightarrow 1]$. A key assumption is that there is ambiguity only as to whether a machine is infected with the attacking malware. This means that for $o \in$: clean, : locked, : offline $p_{obs} = 1$ (i.e. the observer sees the current state) but for $s =$: infected the observer receives observation :infected with probability p_{obs} or :clean with probability $1 - p_{obs}$. The intuition behind this is that some strains of ransomware may initially be stealthy and therefore hard to observe before the ransomware starts to encrypt files. p_{obs} could equivalently be interpreted as the level of competence of malware detection defences.

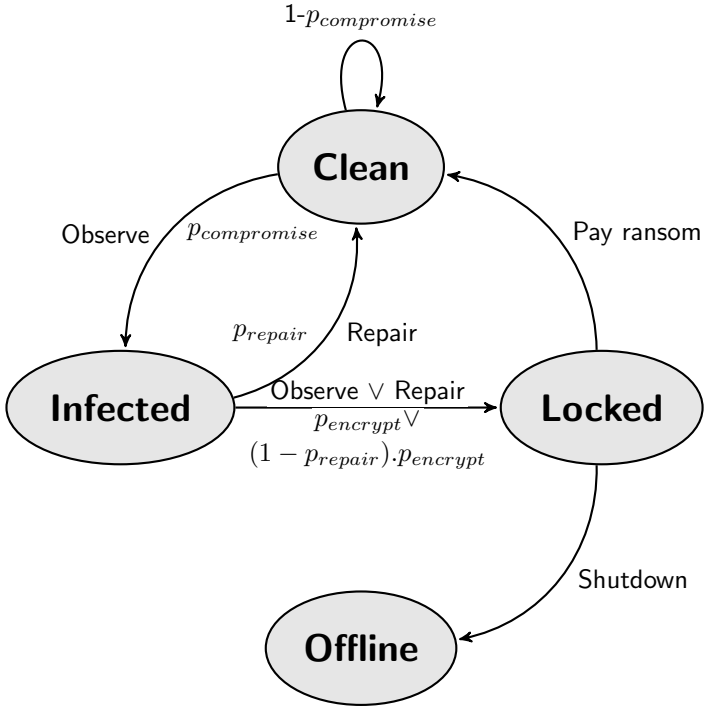
Figure 4 depicts the transition probability structure of the model for a single machine. It should be noted that the actions {Shutdown, Pay Ransom} are deterministic whereas other actions cause the Julia program to return a probability distribution of potential states, which can then be sampled. While the transition structure represents a simplification of the progress of an attack, the aim of the model is to capture the broad dynamics of an attack rather than to model each individual stage intricately.

An important feature of the transition model structure is the two-stage process of ransomware infecting and then encrypting a machine. The justification for this is that in sophisticated ransomware

[5] In Julia, the : prefix denotes a symbol

[6] This could be thought of as analogous to an incubation period in viruses targeting living organisms

Figure 4: Stylised transition probability structure for a single machine



attacks, the attackers may spend time implementing command and control infrastructure and attempting to gain privileges before attempting to launch the ransomware and making demands. It is also not a given that ransomware will prove effective at encrypting data on a given machine. It may be possible for an attacker to launch malicious code on a machine or network, but there is a risk that the code fails to execute as expected due to insufficient privileges, active defences or a combination of measures. This failed attempt would likely be spotted by monitoring personnel, who would then trigger the 'repair' action and attempt to remove the attackers and/or malware from the system.

III.2.2 Expanding the model to a network of machines

The ability to define custom types in *Julia* allows for a ready extension of the single-machine model to a network of machines via a pseudo-objected oriented approach. Each machine within a network is represented by the constructor *NetworkedMachine* with fields:

name, category $\ni \{Workstation, Fileserver, DomainController\}$, boolean initial_vector, boolean isCritical and importance$\ni \{: low, : medium, : high\}$.

The properties of each different type of machine in this specific model are outlined in Figure 5[7]: A

Figure 5: Machine Specification

Machine Category	Name	Initial Vector	Critical	Importance
:workstation	ws	true	false	:low
:fileserver	fs	false	false	:medium
:domaincontroller	dc	false	true	:high

function, *ModelNetwork*, takes as arguments the number of each type of machine, and then constructs a vector of *NetworkedMachines*. Each machine is automatically assigned a name corresponding to the abbreviations in Figure 5 and an integer number. By setting up the problem in this fashion, the POMDP simulations can interact with the network in a manner that is realistic. The inclusion of the initial vector property is particularly important as this allows for fine control of infection modelling with respect to the network topology and privilege structure. For example, the typical initial vectors for ransomware infections are either spear phishing of malware or credential theft for remote access.

[7]These could of course be altered as needed for modelling of a specific use case

In a well-managed network, file servers and domain controllers should not arguably be readily internet facing. This distinction allows the POMDP model to simulate the lateral movement phase of a ransomware attack, which is important for realism.

The states and observations are contained within N-length vectors, **s** and **o**, where N represents the total number of machines in the network. The vectors are ordered in strictly ascending numerical order $dc \rightarrow fs \rightarrow ws$[8]. The set of actions is applied to the system as a whole. There is an argument for having the repair action target individual machines, but it is assumed that a repair action could be scripted and deployed rapidly across the whole network to affected machines (via Powershell or other administrative tools). It is possible that an attacker could attempt to disable this type of administrative control. This is a motivation behind including the probability of repairing an infection failing within the model. However, in an enterprise network, the early stage privileges granted will likely be limited solely to those of the user of that machine who ordinarily should not have such privileges. The available actions are similar to that of the single-machine model: if any machine in the network is observed infected, the repair action becomes available. If a machine is locked, then the shutdown or pay ransom actions become available. In the special case where a domain controller becomes locked, rendering the network unusable, the defender faces an ultimatum of either paying the ransom or shutting down the network.

Within the transitions, it is assumed that once a low importance machine is infected, then the attackers move to infect machines within the network. Separate probabilities are included for low, medium and high importance machines (Figure 6) to allow for different ransomware strategies to be considered. These probabilities are assumed initially to be independent, but this assumption could be relatively easily refined if required for a particular case of interest. There is an argument for considering a network infection model rather than using simple probabilities. However, this would be most justified for a case in which the aim of the attacking malware is to indiscriminately infect as many possible machines and the dynamics of a ransomware attack may be more nuanced. In terms of simulations, the transition probabilities could be parametrised based on the number of infections, but this would add significant complexity to the model.

Figure 6: Network transition probabilities

Probability	
$p_{compromise}$	Probability ransomware initially infects low importance machines
p_{spread_low}	Probability ransomware spreads to other low importance machines
p_{spread_medium}	Probability the ransomware spreads across the network to a medium-importance machine
p_{spread_high}	Probability the ransomware spreads across the network to a high-importance machine
p_{repair}	Probability network cleansed of ransomware before it is locked/files are encrypted
$p_{encrypt}$	Probability that once a machine is infected with ransomware, it becomes locked
p_{obs}	Probability of observations being correct

III.3 Pricing ransomware insurance

As discussed in Section II.6, insurance carriers collect summary data regarding the networks of those looking to purchase cyber-insurance. Realistically, an individual underwriter is likely to have a time constraint in terms of fully evaluating this data. This is especially the case for relatively small policy limits or small/medium enterprise (SME) firms, where a firm may have written thousands of policies or the potential premium intake is modest. The POMDP model presented in this research allows for a representation of a network based on summary data about the number of the machines and is complementary to an underwriting strategy based on mapping specific firm characteristics to past claims. It may also help cyber-insurance firms evaluate policy restrictions - what a firm must do for a claim on an insurance policy to be valid.

Some rudimentary mathematical details of a simple insurance pricing model follow. An insurer writes a policy, $P(p, t, C(\epsilon))$ where p is the premium rate, t is the period of coverage (usually a year),

[8]The vector of states for a single domain controller, single fileserver, three workstation network would thus be [dc1, fs1, ws1, ws2, ws3]

C is the amount of coverage (in monetary units) and ϵ represents the terms of coverage (exclusions, details, sublimits etc.). A policy holder may make claims on losses, l experienced during that year. The insurer will determine whether the claim is valid or not; the policyholder may contest the findings at which point the matter enters the legal rather than purely economic domain. The aim of the insurance company is to ensure that $\sum p_i C_i > \sum l_i$. A rational policyholder will only buy the policy if $pC \leq \sum E[l_i \lambda_i]$ where λ_i is the expected probability of that loss occurring.

The insurance company and buyer compete on information with respect to the decision. The insurance company will have knowledge of the market and risks but the insured may have greater understanding of its own risks. The time dynamic of losses is particularly important for cyber-insurance. In a data breach, costs may be claimed for multiple years after the event, which is problematic for the insurance company who may have by that stage considered the premium intake from the year in question as profit.

In respect of ransomware, for the purposes of this research

$$\epsilon \ni \text{BusinessInterruption}, \text{RansomCosts}, \text{BreachInvestigativeCosts}$$

Within the model, each of these heads of cover has its own separate sub-limit, which will be agreed by the carrier and insured. The POMDP model actions can be mapped to insurance claim states:

$$\begin{aligned} &: \text{shutdown} \rightarrow \text{BusinessInterruption} \\ &: \text{pay} \rightarrow \text{RansomCosts} \\ &: \text{repair} \rightarrow \text{BreachInvestigativeCosts} \end{aligned}$$

One can then run simulations of the POMDP with different reward (cost) values and probabilities with different confidence weightings to aim to derive the optimal premium.

IV Simulations

An initial sensitivity analysis is presented varying different parameters within the model. Two simulations are then introduced: a simple stepwise simulation of the POMDP using three different policies to familiarise the reader with the model structure, and a simulation demonstrating the insurance pricing strategy described in Section III.3. Within these simulations, it is assumed that payment of the ransom restores the system to its original uncompromised state with no risk of reinfection. In reality, this outcome is not guaranteed.

IV.1 Sensitivity analysis

IV.1.1 Varying transition probabilities and network size

The simplest sensitivity analysis is to vary each of the different probabilities within the transaction structure separately, while holding the others constant at $p = 0.5$. The size of the network is initially set at 10 machines, comprising 1 domain controller, 1 fileserver and 8 workstations. This is arbitrary, but seems a reasonable starting point. Separate POMDPs are constructed in *Julia* varying $p = 0.1 \rightarrow 0.9$ in 0.1 step intervals for each probability depicted in Figure 7. The output variable is average number of simulation steps taken until all domain controllers in the network are locked, which effectively represents the problem absolute terminal state. The simulations were run 10,000 times; this value was chosen as it yielded a good balance of convergence and relatively modest computation time ($< 10s$). Unsurprisingly, only varying the probability that the infection spreads to a high importance machine or the probability that once infected a machine is locked have significant bearing on the number of steps for which the simulation runs before reaching a terminal state. This simply verifies that the transition probability structure is operating as designed.

Figure 7: **Sensitivity analysis: number of simulation steps until terminal state reached varying single probability variable, holding others constant at 0.5**

p	$p_{compromise}$	p_{spread_low}	p_{spread_medium}	p_{spread_high}	$p_{encrypt}$
0.1	6	5	5	13	13
0.2	5	5	5	8	8
0.3	5	5	5	6	6
0.4	5	5	5	5	6
0.5	5	5	5	5	5
0.6	5	5	5	4	5
0.7	5	5	5	4	4
0.8	5	5	5	4	4
0.9	5	5	5	4	4

Next, the effect of the size of the network on the number of steps before all high importance machines are locked are investigated. Figure 8 shows the results of this simulation, again with 10,0000 runs. In this simulation, the probabilities are all fixed at a specific value. The transition probabilities determine the ultimate speed with which ransomware can lock a network, so one would expect the number of steps before a terminal state is reached to be inversely proportional to the probabilities. Increasing the network size modestly increases the number of average steps, which a simulation runs.

Finally, the effect on varying the number of domain controllers (i.e. high importance machines) in the network is tested (Figure 9). As in the prior analysis, all transition probabilities are set at value p and the simulations are run 10,000 times. It is found that increasing the number of domain controllers in the network generally increases the amount of steps before a terminal state is reached except for at extremely high probabilities of infection/spread/encryption. This suggests that for a network with reasonable defences, there is economic benefit to having multiple domain controllers, perhaps in a failsafe-type configuration.

Figure 8: **Sensitivity analysis: number of simulation steps until terminal state reached varying all probabilities to p with different network sizes.**

	#(high, medium, low) importance machines				
p	(1,1,8)	(2,2,16)	(3,3,24)	(4,4,32)	(5,5,40)
0.1	22	28	33	36	38
0.2	11	14	16	18	19
0.3	8	10	11	12	13
0.4	6	7	8	9	10
0.5	5	6	7	7	8
0.6	4	5	6	6	6
0.7	4	4	5	5	5
0.8	4	4	4	4	4
0.9	3	3	3	3	4

Figure 9: **Sensitivity analysis: number of simulation steps until terminal state reached varying number of domain controllers in network, 5 fileservers, 100 workstations**

	# Domain Controllers				
p	1	2	3	4	5
0.1	21	28	33	36	38
0.2	11	14	16	18	19
0.3	8	10	11	12	13
0.4	6	7	8	9	9
0.5	5	6	7	7	8
0.6	4	5	6	6	6
0.7	4	4	5	5	5
0.8	4	4	4	4	5
0.9	3	3	3	4	4

IV.1.2 Effect of transition probability variation on utility

The next simulation run is to check how the utility parameters evolve as an infection spreads without intervention (i.e. the POMDP action is held at :observe). A network of 5 domain controllers, 5 fileservers and 40 workstations is used. This is an arbitrary choice but using a large network allows for variation to be more readily observed as demonstrated by the results in Figure 8

The utility components are defined as follows:

- C: 1 - (%medium and high importance machines infected or locked)
- I: 1 - (%machines infected)
- A: 1 - (%machines locked)

First $p_{encrypt}$ is varied, holding all other probabilities constant at 0.5 (Figure 10). As expected, because availability is solely a function of encryption, there is divergence only in this parameter. This provides a useful test that the simulations are running as expected. Next, all probabilities are held constant except for $p_{compromise}$ and p_{spread_low}, which should have an effect particularly on integrity and availability (in this simulation, there is a 50% chance that an infected machine become locked in the following step). As shown in Figure 11, there is no variation in confidentiality but both integrity and availability decrease rapidly as a function of $p_{compromise}$ and p_{spread_low}.

Finally, p_{spread_medium} and p_{spread_high} are varied (Figure 12). This has the largest impact on confidentiality as expected given its definition, but also some impact on integrity and availability though to a lesser extent given that there are 40 low importance machines in the sample network but only 5 medium and 5 high importance ones.

Figure 10: Utility versus number of simulation steps, holding all probabilities constant at 0.5 except $p_{encrypt}$, which varies per the figure legend.

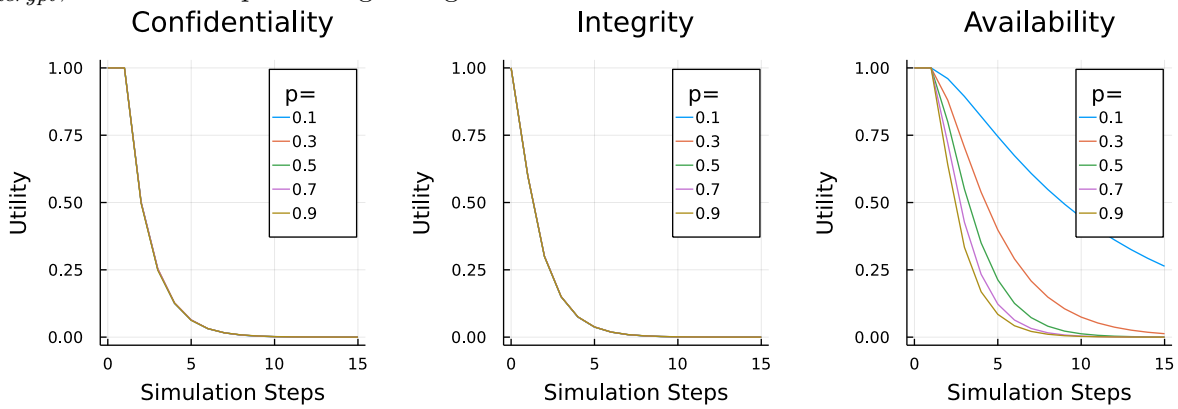
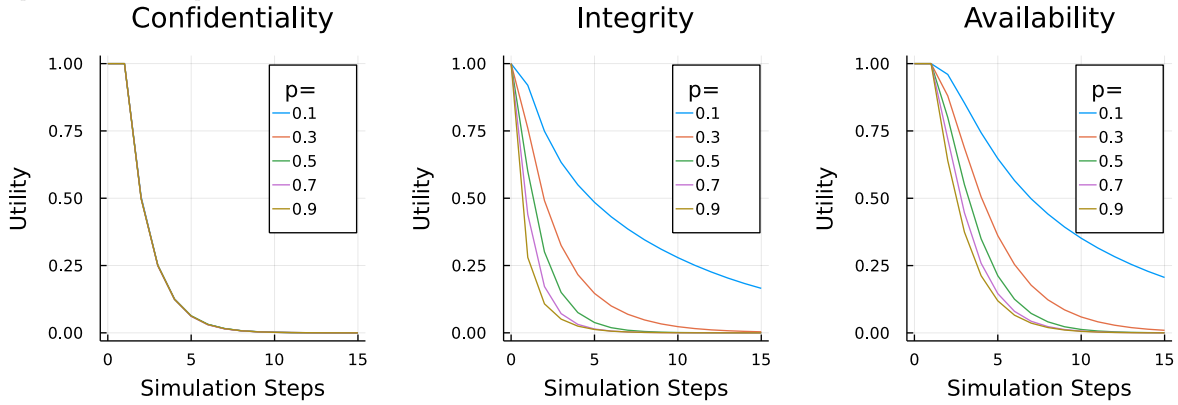


Figure 11: Utility versus number of simulation steps, holding all probabilities constant at 0.5 except $p_{compromise}$ and p_{spread_low}, which vary per the figure legend.



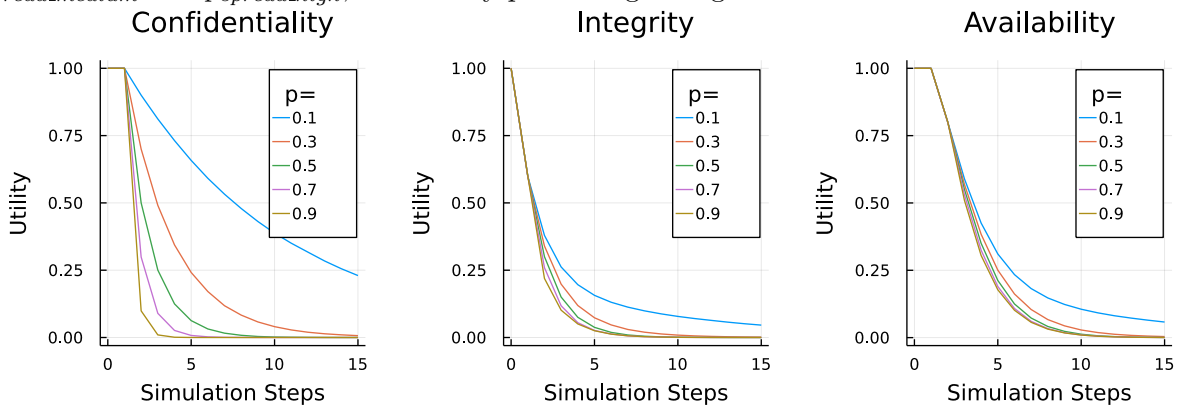
These results are designed to illustrate how simple utility metrics can be used to gain a picture of the evolution of a moderately complex and uncertain simulation and the variation of key parameters. Such plots could be used to simulate the impact of complex technical defences and present the results to non-technical key decision makers. The subsequent simulations in this paper will demonstrate some applied cases of the effect of different defence strategies against various strains of ransomware with divergent characteristics.

IV.2 Ransomware infection

IV.2.1 Specification

This simulation considers defence against three different ransomware strain attack scenarios: Rare/Sophisticated, Common/Unsophisticated and 50-50 Baseline (see Figure 13 for a full specification). The first of these is designed to replicate a targeted strain of highly effective ransomware, which is not commonly observed in the wild but once inside a system proves very effective at facilitating lateral movement and ultimately conferral of domain administrator privileges. The probability of initial infection is set relatively low at 0.1, but if a low importance machine is compromised then it spreads quickly to other machines. The probability of successfully infecting medium and high importance machines are set lower at 0.6 and 0.5 respectively to reflect the fact that these servers may be actively monitored and likely have some more sophisticated defences and/or policies aimed to prevent them being susceptible to malicious activity. $p_{encrypt} = 0.7$ for this strain. The second strain studied is commonly observed, self-propagating ransomware such as WannaCry, which is readily eliminated by appropriate tools. Here, the probability of initial infection is set at a very high 0.9, but the probability of repair is set at 0.8; thus there is a decent, but not certain, chance that this strain might be cleared

Figure 12: Utility versus number of simulation steps, holding all probabilities constant at 0.5 except p_{spread_medium} and p_{spread_high}, which vary per the figure legend.



from any workstation it infects. It is assumed that its locking/encrypting methodology is not that sophisticated, expressed by $p_{encrypt} = 0.3$. Finally, as the name suggests, the 50-50 baseline scenario sets all probabilities in the model to 0.5. Observation accuracy is set at 70% initially to create the possibility of inaccurate observations and consequent policy errors. For completeness, a simple discount factor of 0.95 is set, though this is not required for simulations but would be used if applying a solver to the system.

Figure 13: Simulation Specifications

	Rare/Sophisticated	Common/Unsophisticated	50-50
Rewards			
$r_{observe}$	-1	-1	-1
r^{+}_{repair}	-2	-2	-2
r^{-}_{repair}	-10	-10	-10
$r_{shutdown}$	-150	-150	-150
r_{ransom}	-50	-50	-50
Probabilities			
$p_{infection}$	0.1	0.9	0.5
p_{spread_low}	0.8	0.3	0.5
p_{spread_medium}	0.6	0.1	0.5
p_{spread_high}	0.5	0.1	0.5
p_{repair}	0.2	0.8	0.5
$p_{encrypt}$	0.7	0.3	0.5
Other			
$Obs.Acc.$	0.7	0.7	0.7
$Disc.Fac.$	0.95	0.95	0.95

The reward parameters selected are intended to be largely illustrative and are arguably the most transparent component of the model. There is a small penalty for observation, which is designed to represent the cost of monitoring a network. Separate rewards are included for successful and failed repairs (r^{+}_{repair} and r^{-}_{repair} respectively). Intuitively, an unsuccessful repair means likely further investigative costs or expense to attempt to remove the ransomware for the network, such as hiring specialist help. The costs of shutting down the network are deliberately set as higher (i.e. more negative in reward terms) than paying the ransom. The aim of this simulation is to investigate how defensive actions affect the resultant outcomes and consequently, the rewards are simply a means of 'keeping score'. While abstract in relation to real world costs, this approach is consistent with conventions within the game theory and decision model literature.

The three strains are tested on a sample network containing 1 domain controller, 2 fileservers and 10 workstations. This network size was chosen to provide a reasonably sized attack surface but to be

of a manageable size for debugging purposes. When aiming to solve, or at least simulate a POMDP, it is conventional to evaluate the effect of different policies. A policy in this context is a specification of actions corresponding to a belief (in this model, the belief is simply the observations). Three policies are evaluated: the 'cautious policy', the 'gambler policy' and the 'random policy'.

- **Cautious policy**: attempt repair if infected; shut down if domain controller encrypted/locked; never pay ransom.
- **Gambler policy**: observe until a system becomes encrypted/locked at which point pay ransom.
- **Random policy**: take random action from set of available actions corresponding to received observation.

The simulations are run in step-wise fashion:

1. Initial vector of states s and observations o set fully clean
2. Receive optimal action a from policy p based on o
3. Determine next state s' from transition $t(s, a)$
4. Compute reward $r(s, a, s')$
5. Record s, a, s', r
6. If action is terminal, terminate simulation
7. Set $s = s'$ and compute observations $o(s)$
8. Repeat from (2) until maximum number of steps reached or terminal action taken

IV.2.2 Results

For each POMDP and policy, the simulations were run 10,000 times and the history recorded. A maximum of 15 steps was permitted in each individual simulation - per Figure 8, this is likely to be sufficient to fully capture the simulation steps in most outcomes. As expected, the average reward

Figure 14: Simulation Results - average wealth

	Rare/Sophisticated	Common/Unsophisticated	50-50
Cautious Policy	-146	-68	-136
Gambler Policy	-53	-52	-52
Random Policy	-111	-112	-111

(Figure 14) for the cautious policy is much lower than the gambler policy, given that the cautious policy prohibits ransom payment and the cost of shutting down the system is higher than the ransom. This is particularly apparent for the rare but dangerous strain of ransomware. However, for the common but benign strain, as it is far less likely that the key network infrastructure is locked, the difference between average rewards is much smaller.

Figure 15: Simulation step distribution for rare/sophisticated strain

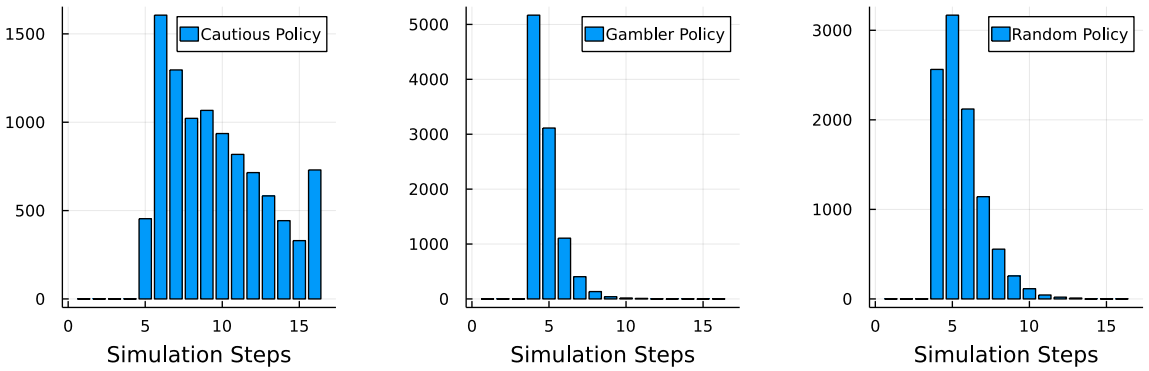


Figure 16: Simulation step distribution for common/unsophisticated strain

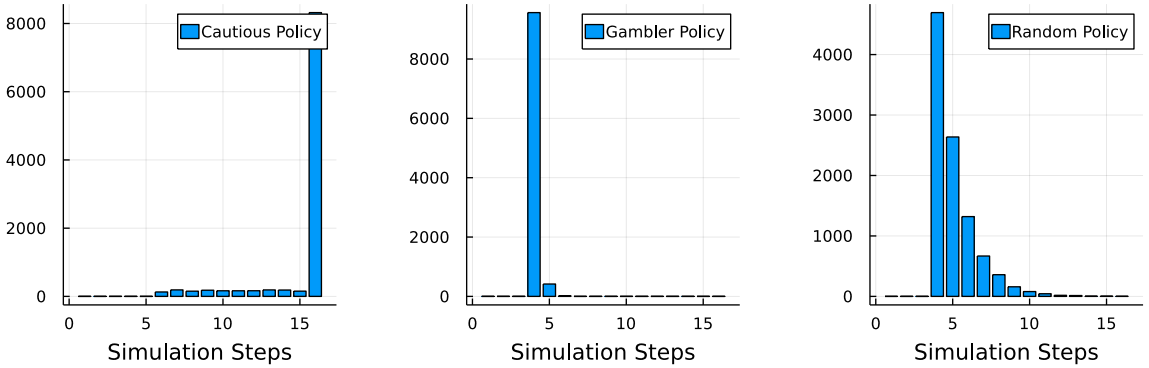


Figure 17: Simulation step distribution for 50/50 strain

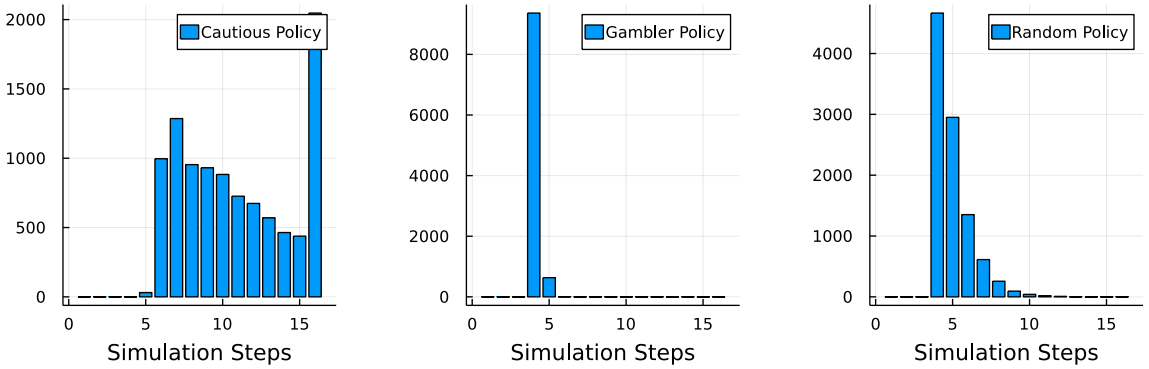


Figure 18: Utility evolution for rare/sophisticated strain

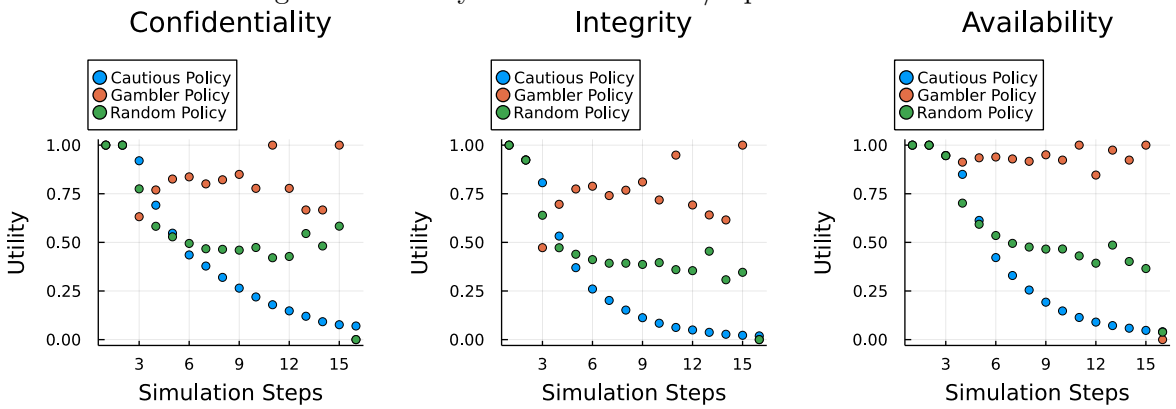


Figure 19: Utility evolution for common/unsophisticated strain

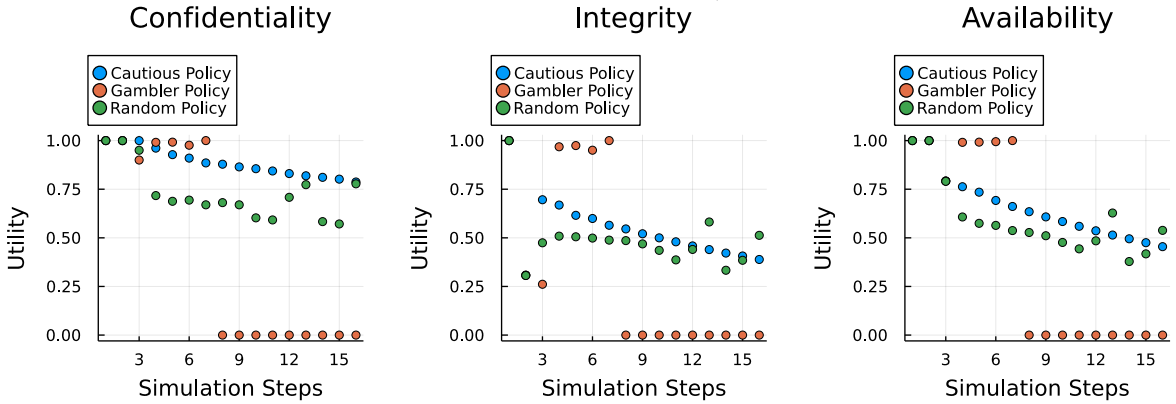
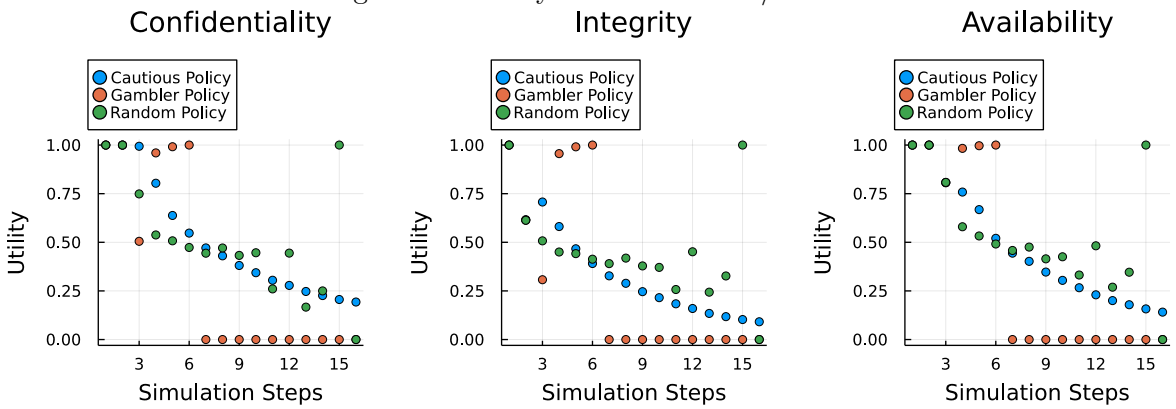


Figure 20: Utility evolution for 50/50 strain



Figures 15, 16, 17 show the distribution of the number of simulation steps before the simulation terminates across the 10,000 runs. This provides an insight into the variability of the length of the simulation and the impact of the policy chosen. For the rare/sophisticated ransomware strain, the cautious policy shows the greatest variability. This is likely because the probability of repair is low, and thus the chances of the domain controller becoming locked are relatively high, as suggested by the average reward returned being close to the cost of shutting down the system. The gambler policy in contrast results in much shorter run times. For the common/unsophisticated strain, there is almost a deterministic distribution of outcomes, which makes sense given that the probability of a repair is much higher than the probability of the infection spreading.

The utility for each state in the simulations was calculated (as in Section IV.1.2). Figures 18, 19 and 20 show the average components of the utility for each ransomware strain and each policy. The x-axis of each subplot represents the number of simulation steps and the y-axis the numerical utility, ranging from 0 to 1. For each of the 10,000 runs, the number of simulation steps taken was recorded and transformed into a vector so that the average is correctly calculated. For the rare/dangerous strain, the gambler policy maximises utility whereas the cautious policy drastically underperforms the benchmark random policy. For the common/unsophisticated strain, however, the cautious policy performs notably better, with the domain controller locked in only 25% of simulations and on average fewer than 50% of network machines either infected with ransomware or locked. The 50/50 strain is intended as a control; the gambler policy has notably less potential for randomness in the outcomes whereas in the cautious policy, the repair action proves ineffective at stemming the spread and progress of the ransomware.

It is relatively straightforward to assess the effect of varying the parameters within the different POMDPs on the average rewards received for the different policies. Figure 21 shows the average reward for the cautious/unsophisticated strain POMDP varying the probability that an infection spreads to the domain controller once in the system (p_{spread_high}). This illustrates that the crossover point between the gambler policy being the optimal strategy that the cautious policy occurs at a fairly low probability of overall domain controller locking. This is largely because the gambler policy immediately pays the ransom thus preventing the infection from spreading to the domain controller.

A useful experiment is to investigate the effects of varying the probability of a repair being successful on the reward (Figure 22) for the common/unsophisticated ransomware strain where this probability should have greatest impact. This provides a useful check as to the robustness of the policies as only the cautious policy reward should vary with p_{repair}. While in a real world decision, the evaluation of defences would not be undertaken purely on the basis of probabilities, this nevertheless illustrates the sort of cost-benefit analysis that might be undertaken when planning security investments.

Figure 21: Varying probabilities of domain controller compromise, Common/Unsophisticated Strain

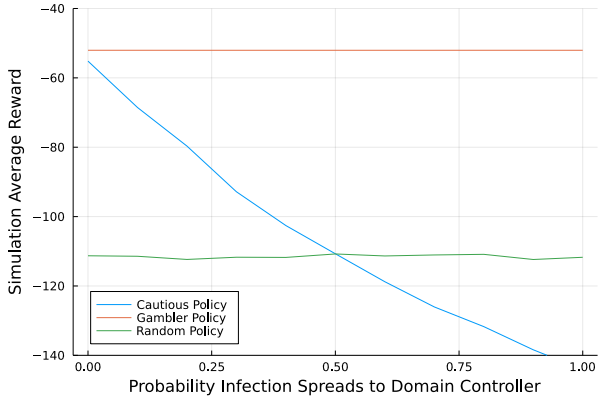
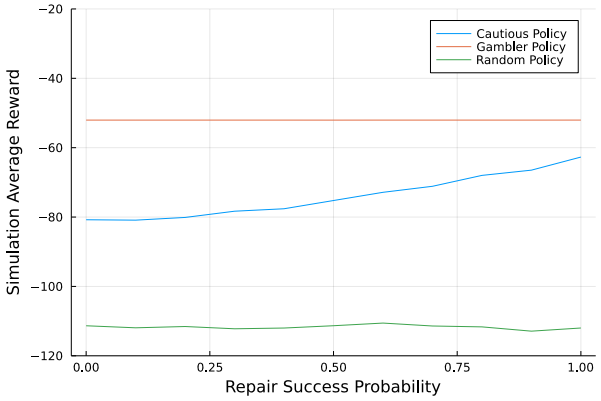


Figure 22: Varying probabilities of successful repair, Common/Unsophisticated Strain



IV.3 Insurance Pricing

This section considers a simple insurance pricing example against ransomware based on the POMDP model introduced. An insurer is considering pricing ransomware insurance for an organisation. The organisation has a network of 1,000 workstations, 10 fileservers and 5 domain controllers. The policy has the following features:

- Ransom paid up to $1mn only if all domain controllers locked.
- Repair costs paid. A successful repair costs $10,000 and an unsuccessful one $50,000 each time.
- Business interruption costs of $10mn in the event of more than 50% machines in network locked by ransomware.

For simplicity in this example, these costs are calculated for 10 different ransomware attack scenarios, setting

$$p = p_{infection} = p_{spread\_low} = p_{spread\_medium} = p_{spread\_high} = p_{encrypt} = 1 - p_{repair} \tag{2}$$

This means that as ransomware becomes more efficient and effective, the probability of repairs being successful decreases commensurately. The POMDP simulation can then be run for

$$p = 0.1, 0.2, \ldots, 0.9 \tag{3}$$

and the average reward then recorded. Simulations terminate once either a ransom is paid, all machines are cleaned or the business interruption threshold is reached. An initial run of 1000 iterations with a maximum of 20 steps per simulation yields the following average costs:

Figure 23: Simulation Results varying probabilities

p	Loss	Weighting
0.1	$703,200	0.3
0.2	$764,750	0.2
0.3	$800,000	0.1
0.4	$8.68mn	0.1
0.5	$10.35mn	0.1
0.6	$10.32mn	0.05
0.7	$10.09mn	0.05
0.8	$10.05mn	0.05
0.9	$10.05mn	0.05

The insurer could then apply its own analysis (or source it from a third party) regarding the ransomware environment and generate a distribution of possible outcomes. These are shown as weightings in Figure 23; the values provided here are purely illustrative. Per Section III.3 the simulation gives an average expected loss for the scenario of $4.38mn. This is a relatively abstract scenario, but is designed purely as a proof of concept to demonstrate how POMDP models could contribute to insurance pricing. Such modelling is likely to represent a useful direction for further subsequent work using organisations with difference network architectures and parameters.

V Further Work

The simulation demonstrates a proof-of-concept of a POMDP approach to modelling ransomware. Ideally, the next steps in the work would be to use the framework to evaluate decision making in specific scenarios and systems architecture and feedback is welcomed as how this might be most usefully achieved. The representation of the network was constructed with the aim of replicating sample networks such as an Active Directory network and the demonstration of its usage in this work

is fairly simple. A potentially interesting expansion of the simple approach would be to introduce labelled transitions to formally describe the privilege structure between machines. This might allow for incorporating user accounts and privilege structures within the communications and may be of use in threat modelling to describe various different potential attack vectors from unintended use of privileges (for example from compromise of service account credentials). It should be noted that the construction of the model potentially allows for separate model networks to be constructed, representing an Active Directory Forest, for example.

There is the potential to introduce significant complexity into models such as the one presented in this work. For simplicity, it is assumed that a ransom payment results in full decryption and restoration of the system to its original state. This may not be the case in reality and there would be potential scope to incorporate this into future model simulations. Equally, it is assumed that once a machine is infected, if not repaired, it is encrypted or locked with fixed probability. If an attacker is able to gain introduce command and control (C2C) functionality, then this might not be the case.

The model presented within this research focused on a single POMDP and assumes no costs to the criminal actor. An interesting expansion of the model may be to simulate such costs on the criminal actor (for example, resource constraints, risk of discovery within a network etc.). The criminal actor might also be simulated as a reinforcement learning (RL) agent; one could also potentially introduce a defender (RL) agent as well.

VI Conclusion

This paper has introduced a POMDP model for simulating ransomware attacks either on a single machine or a network of machines of varying importance. The results of a simple simulation of different types of ransomware attack highlight that economically the least costly financial outcome is usually to pay the ransom at the first chance, although this is a scenario that is unlikely to be encouraged by governmental authorities or insurers. It is hoped that this model may be useful for helping frame simulations of complex attacks and in developing optimal defence strategies. The applicability of such model results to a simple insurance pricing example has also been demonstrated, highlighting how cover could be adapted based on risk perception.

VII Acknowledgements

David Pym and Christos Ioannidis at UCL provided helpful constructive criticism at various stages of the research described in this paper. Thanks are due to my colleagues on the UCL Centre for Doctoral Training programme in Cybersecurity for their encouragement, friendship and willingness to patiently listen to new ideas. The work was inspired by an introductory talk on POMDPs by Mykel Kochenderfer of Stanford University for the PPLV Group at UCL. Daniel Woods and Julian Williams provided useful comments in the early stages of the work, while the three WEIS reviewers proposed suggestions that have greatly improved the paper. This work was supported, in part, by the Engineering and Physical Sciences Research Council grant for Doctoral Training EP/R513143/1.

VIII Appendix

VIII.1 Summary of model parameters

The following figures are included within the text of the paper but are included here together for convenience.

VIII.1.1 Inputs

Figure 24: Summary of inputs

Parameter	Description
Probabilities	
$p_{compromise}$	Probability ransomware initially infects a low importance machine
p_{spread_low}	Probability ransomware spreads to other low importance machines
p_{spread_medium}	Probability the ransomware spreads across the network to a medium-importance machine
p_{spread_high}	Probability the ransomware spreads across the network to a high-importance machine
p_{repair}	Probability a machine is cleansed of ransomware before it is locked/files are encrypted
$p_{encrypt}$	Probability that once a machine is infected with ransomware, it becomes locked
p_{obs}	Probability of observations being correct
Rewards	
$r_{observe}$	Cost of observation
r^{+}_{repair}	Cost of successful repair
r^{-}_{repair}	Cost of unsuccessful repair
$r_{shutdown}$	Cost of shutting down system
r_{ransom}	Cost of ransom payment

VIII.1.2 States

The state of the network is represented as an ordered vector, **s**, of symbols. $s_i = :\text{clean}$ for all i initially, with each s_i updated as the simulation proceeds. The ordering is determined by $i = [dc_1, \ldots, dc_n, fs_1, \ldots, fs_n, ws_1, \ldots, ws_n]$ where dc_n, fs_n, ws_n are the numbers of domain controllers, fileservers and workstations respectively in the network.

VIII.1.3 Actions

Figure 25: Model actions

State	Observe	Repair	Shutdown	Pay
Clean	x	(x)		
Infected	x	x		
Locked			x	x
Offline				

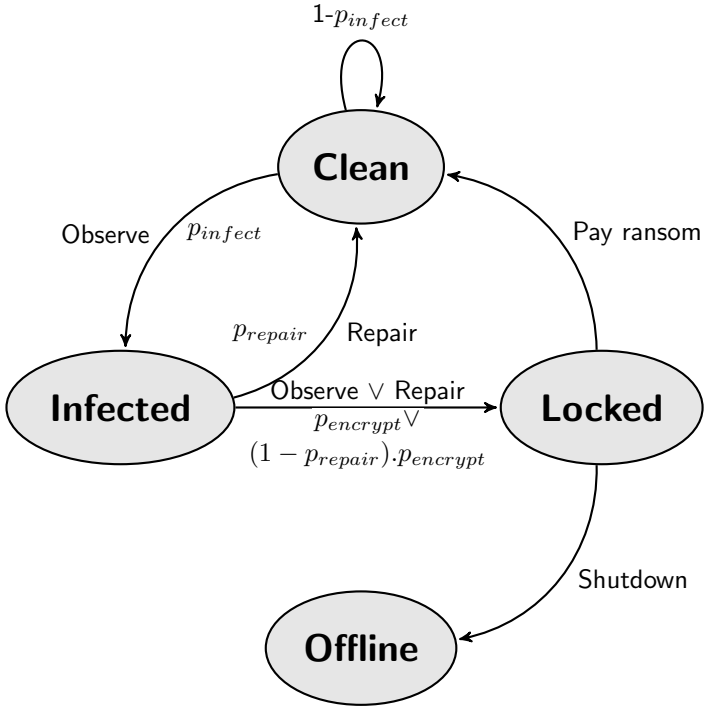
VIII.1.4 Network machine parameters

Figure 26: Machine Specification

Machine Category	Name	Initial Vector	Critical	Importance
:workstation	ws	true	false	:low
:fileserver	fs	false	false	:medium
:domaincontroller	dc	false	true	:high

VIII.1.5 Transition model structure

Figure 27: Stylised transition probability structure



$p_{infect} = p_{compromise}|p_{spread\_low}|p_{spread\_medium}|p_{spread\_high}$ depending on the machine type and current system infection status.

VIII.1.6 Utility

- Confidentiality: 1 - (%medium and high importance machines infected or locked)
- Integrity: 1 - (%machines infected)
- Availability: 1 - (%machines locked)

References

[1] A. Young and Moti Yung. "Cryptovirology: extortion-based security threats and countermeasures". In: *Proceedings 1996 IEEE Symposium on Security and Privacy*. 1996, pp. 129–140. DOI: 10.1109/SECPRI.1996.502676.

[2] David S. Wall. "The Internet as a Conduit for Criminal Activity". In: *Information Technology and the Criminal Justice System*. Ed. by A Pattavina. 2015th ed. Thousand Oaks, California: Sage Publications, 2005, pp. 77–98. URL: https://papers.ssrn.com/abstract=740626 (visited on 04/29/2020).

[3] Osterman Research. *How to Reduce the Risk of Phishing and Ransomware*. White Paper.

[4] Andy Greenberg. *The Untold Story of NotPetya, the Most Devastating Cyberattack in History*. URL: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/.

[5] Aron Laszka, Sadegh Farhang, and Jens Grossklags. "On the economics of ransomware". In: *International Conference on Decision and Game Theory for Security*. Springer. 2017, pp. 397–417.

[6] Terrence August, Duy Dao, and Marius Florin Niculescu. "Economics of ransomware attacks". In: *Earlier Version Presented at WEIS* (2017).

[7] Edward Cartwright, Julio Hernandez Castro, and Anna Cartwright. "To pay or not: game theoretic models of ransomware". In: *Journal of Cybersecurity* 5.1 (2019), tyz009.

[8] Zhen Li and Qi Liao. "Ransomware 2.0: to sell, or not to sell a game-theoretical model of data-selling Ransomware". In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*. 2020, pp. 1–9.

[9] Pierce Ryan et al. *Dynamics of targeted ransomware negotiations*. 2021. arXiv: 2110.00362 [math.DS].

[10] Erick Galinkin. "Winning the Ransomware Lottery". In: *International Conference on Decision and Game Theory for Security*. Springer. 2021, pp. 195–207.

[11] Tongxin Yin, Armin Sarabi, and Mingyan Liu. "Deterrence, Backup, or Insurance: A Game-Theoretic Analysis of Ransomware". In: Workshop on the Economics of Information Security. URL: https://weis2021.econinfosec.org/wp-content/uploads/sites/9/2021/06/weis21-yin.pdf.

[12] Carlos Sarraute, Olivier Buffet, and Jörg Hoffmann. "POMDPs make better hackers: Accounting for uncertainty in penetration testing". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 26. 1. 2012.

[13] Carlos Sarraute, Olivier Buffet, and Jörg Hoffmann. "Penetration testing== POMDP solving?" In: *arXiv preprint arXiv:1306.4714* (2013).

[14] Jörg Hoffmann. "Simulated Penetration Testing: From" Dijkstra" to" Turing Test++"". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 25. 1. 2015.

[15] Vineet Mehta et al. "Decision-theoretic approach to designing cyber resilient systems". In: *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2016, pp. 302–309.

[16] Mohamed C Ghanem and Thomas M Chen. "Reinforcement learning for efficient network penetration testing". In: *Information* 11.1 (2020), p. 6.

[17] Jonathon Schwartz, Hanna Kurniawati, and Edwin El-Mahassni. "POMDP+ Information-Decay: Incorporating Defender's Behaviour in Autonomous Penetration Testing". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 30. 2020, pp. 235–243.

[18] Daniel Woods and Rainer Boehme. "How Cyber Insurance Shapes Incident Response: A Mixed Methods Study". In: Workshop on the Economics of Information Security. URL: https://weis2021.econinfosec.org/wp-content/uploads/sites/9/2021/06/weis21-woods.pdf.

[19] Damian Glynn, Sue Taylor, and Steven Nock. *The Basic Business Interruption Book*. 2020. URL: https://www.cila.co.uk/cila/download-link/sig-downloads/business-interruptions/371-cila-the-basic-business-interruption-book-2020/file.

[20] Burak Filiz et al. "On the Effectiveness of Ransomware Decryption Tools". In: *Computers and Security* 111 (2021), p. 102469. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102469. URL: https://www.sciencedirect.com/science/article/pii/S0167404821002935.

[21] Grégoire Jacob, Eric Filiol, and Hervé Debar. "Malware as interaction machines: a new framework for behavior modelling". In: *Journal in Computer Virology* 4.3 (2008), pp. 235–250.

[22] Mila Dalla Preda and Cinzia Di Giusto. "Hunting Distributed Malware with the κ-Calculus". In: vol. 6914. Aug. 2011, pp. 102–113. ISBN: 978-3-642-22952-7. DOI: 10.1007/978-3-642-22953-4_9.

[23] Hasan Cam. "Online detection and control of malware infected assets". In: *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*. 2017, pp. 701–706. DOI: 10.1109/MILCOM.2017.8170869.

[24] Wanping Liu. "Modeling Ransomware Spreading by a Dynamic Node-Level Method". In: *IEEE Access* 7 (2019), pp. 142224–142232. DOI: 10.1109/ACCESS.2019.2941021.

[25] Zhisheng Hu, Minghui Zhu, and Peng Liu. "Adaptive Cyber Defense Against Multi-Stage Attacks Using Learning-Based POMDP". In: *ACM Trans. Priv. Secur.* 24.1 (Nov. 2020). ISSN: 2471-2566. DOI: 10.1145/3418897. URL: https://doi.org/10.1145/3418897.

[26] Daniel Woods et al. "Mapping the coverage of security controls in cyber insurance proposal forms". In: *Journal of Internet Services and Applications* 8.1 (2017), pp. 1–13.

[27] Sasha Romanosky et al. "Content analysis of cyber insurance policies: How do carriers price cyber risk?" In: *Journal of Cybersecurity* 5.1 (2019), tyz002.

[28] RC Jason Nurse et al. "The data that drives cyber insurance: A study into the underwriting and claims processes". In: *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. IEEE. 2020, pp. 1–8.

[29] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. "Planning and acting in partially observable stochastic domains". In: *Artificial intelligence* 101.1-2 (1998), pp. 99–134.

[30] Maxim Egorov et al. "POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty". In: *Journal of Machine Learning Research* 18.26 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-300.html.