Assessing The Risk Of AI-Enabled Computer Worms

Authors: John Halstead & Luca Righetti | Last Updated: Sep 10, 2025

Executive Summary

AI has been used to support many aspects of cyberdefence and cyberoffence for some time. However, in light of recent rapid progress in AI, some experts have expressed concern that large and sudden improvements in future AI-cyber capabilities could pose severe risks if not managed properly.

These concerns have prompted several frontier AI companies to define offensive cyber capability thresholds in their safety and security policies. These policies state that: *if* an AI model reaches a certain level of cyber capability, *then* the company should not release the system until it has appropriately mitigated the risk. In turn, companies have built several AI-cyber benchmarks and "red teaming" exercises to test whether models trigger these thresholds.

However, at present, there is a lack of published *AI-cyber threat models*. Threat models are evidence-based analyses of how much AI capabilities increase risk: for example, a threat model would estimate the economic costs if AI showed strong performance on vulnerability discovery or malware development. Due to the lack of published threat models, AI companies' cyber capability thresholds often lack clear justifications and diverge substantially from one another. It also limits policymakers' foresight into emerging cyber risks, and makes it difficult to know whether current cyber model evaluation results warrant concern or not.

This report aims to help bridge the gap between model evaluations and estimates of societal risk by reviewing one AI-cyber threat model in depth. We focus specifically on how AI-enabled discovery of *critical software vulnerabilities* and development of powerful *exploits* of them could increase the risk of computer worms that damage data on a large number of devices. Historically, computer worms have been amongst the most damaging cyber risks.

We use multiple sources of evidence, including qualitative case studies, subject-matter expert interviews, and existing cyber risk scenarios. To help inform our overall forecast we develop a simple model of computer worm risk and estimate that the baseline risk (i.e. assuming no further improvements in AI) of data damaging worms is $150M/yr (with a 95% credible range of $6M-$4B). Conditional on a hypothetical scenario in which there is a *very* strong and sudden improvement in AI-enabled vulnerability discovery and exploit development, the risk increases by a factor of 10 to around $1B/yr. To validate this result, we conducted a small pilot expert survey independent of this analysis. Survey respondents estimated that the baseline risk is higher – closer to $1B/yr – and that AI uplift would increase the risk to $3B to $8B per year.

It is unclear whether risks of this magnitude count as catastrophic enough to justify management via AI Safety Policies. For example, OpenAI's Preparedness Framework aims to mitigate AI capabilities that could lead to hundreds of billions of dollars of economic damage. If our estimates are correct, the computer worms risk scenario described does not meet OpenAI's threshold. However, it is unclear what risk thresholds many other AI companies use, and the question of which risk thresholds ought to be used is under-researched. Companies also have reason to take action to mitigate risks that don't meet this high bar, outside of formal Safety Policies. Moreover, elite exploit discovery might cause harm via other pathways, such as industrial or state espionage, so this threat model is a lower bound on the potential damages from this AI capability. Overall, it is unclear how AI companies ought to respond to this threat model. We hope the methods and analysis in this report can provide a template for further work in the space.

Computer worms have been among the most damaging cyber events, making them a plausible candidate threat model

Prior to this report, we reviewed 34 case studies of major historical cyberattacks, including state espionage, data theft, critical infrastructure attacks, worm attacks, and so on. For most types of cyberattack, our analysis tentatively suggests that even significant increases in AI capabilities would be unlikely to cause billions of dollars of economic damage. For many types of cyberattack, there is a lack of precedent for close-to-catastrophic harm, and AIs would have to surpass human performance across a diverse range of agentic tasks in order to increase the risk.

Compared to many other types of cyberattack, AI-enabled worms are a more concerning source of risk. Three key findings, which this report presents at greater length, are that:

1. **There is precedent for data damaging worms causing billions of dollars of damage — and reason to believe greater harm is possible.** In 2017, for example, two worms known as WannaCry and NotPetya caused approximately ~$1B and ~$10B in damage respectively in less than a day. The harm could have been even worse: the WannaCry code contained an error that allowed defenders to prevent further spread and damage, while NotPetya was designed to restrict damage mainly to Ukraine.
2. **Many actors are willing to launch worms**: Prior to 2005, individual hackers regularly launched worm attacks that went on to infect thousands to tens of millions of systems. Since then, due to improved cybersecurity, major worm attacks have become much less frequent and have been carried out by more sophisticated actors. This suggests that many actors would launch worms if AI enabled them to do so.
3. **Worm attacks are bottlenecked by just a few narrow technical skills: vulnerability discovery and developing what we call "elite exploits"[1] of these vulnerabilities.** For

[1] We define "elite exploits" as exploits that: (1) are "zero-click": require no action on the part of the user in order to spread; (2) allow remote code execution: allow attackers to remotely to execute arbitrary code on

> example, both WannaCry and NotPetya were enabled by a leaked elite exploit initially developed by the NSA and seemed to require much lower sophistication after this point. This contrasts other types of cyber attacks that have hard steps spread across the attack chain and rely on different skills.. If future AI capabilities lower the bar to developing elite exploits, then this by itself could meaningfully raise the likelihood of worm attacks.

Vulnerability discovery and exploitation is already a major focus for AI model evaluations. However, we want to emphasise that the exploits we focus on here are much harder to develop than those typically included in model evaluations, and frontier models currently fall well-short of being able to find elite exploits. Nonetheless, model capabilities are improving rapidly, and it is difficult to rule out large jumps in model capabilities in the future.

Estimates of the effect of AI capabilities on risk can be produced using empirically-grounded threat models

This report aims to generate estimates of how risks from data damaging worms could increase if AI systems enable different threat actors to find elite exploits. Our approach is to outline an empirically-grounded threat model that decomposes the question into multiple parameters, which can then be filled in to produce an overall forecast of the baseline risk (absent further improvements in AI), and the additional risk created *if* AI reaches a given capability level in the future.[2]

We combine multiple lines of evidence to inform the threat model and our own parameter estimates. These lines of evidence include: historical case studies, analysis of economic damage estimates, and feedback from cyber experts. The model results are necessarily uncertain, but simple models have the advantage of making key assumptions transparent. Since any single assessment risks being subjective, we also surveyed 8 subject-matter-experts and 13 superforecasters to produce their own risk estimates. Given the small sample, the results of the survey should be interpreted with caution, but it provides initial information on a wider range of perspectives. Future work could expand the survey to produce more robust consensus estimates. We will discuss the full survey results and methodology in forthcoming work.

Results

The author's estimates are summarised in Table ES1. The table shows the risk posed by five different classes of threat actors, ranging from individual hobbyist hackers to the world's most

a system without the user's knowledge; (3) have admin or higher privileges: privileges are the permissions granted to users, programs, or processes to perform specific actions on a system or access particular resources and range from (low) sandbox application privileges to (high) system-level privileges; and (4) are effective against >10M systems.

[2] To structure our analysis, we broke our damage estimate down into several parameters: a threat actor's *capability* to develop elite exploits and data damaging worms, their *willingness* to launch worm attacks if able, and the likely *damages* from successful attacks. We applied this to five different categories of threat actor, ranging from individual hobbyist hackers up to teams of more than 100 state-sponsored experts.

capable states. The findings from the author’s own estimates and the pilot survey are summarised below:

- **Baseline damages**: The author estimated baseline damages (i.e. assuming no further progress in AI) from data damaging worm attacks of $150M [$6M-$4B]. Almost all of this risk is driven by more sophisticated actors consisting of teams of state-level hackers
- **Risk Scenario:** To operationalise the effect of a large and sudden increase in AI cyber capabilities, we hypothesised that a study finds that a future AI system enabled 25% of individual professional hackers to develop elite exploits with three months of full time effort – and that this model was immediately released as open-sourced without additional safeguards.
- **Additional AI-enabled damages:** The author estimates that *if* this scenario were to happen, *expected* social damages (the social damages that could occur, weighted by their probability) would increase by ~10X to around $1B. Most of the increased risk comes from uplift to individual professional hackers, many of whom would be willing to launch worm attacks, but currently lack the ability to do so.
- **Pilot Survey:** Expert estimates implied baseline damages of $6B [$500M-$83B], and the superforecaster estimates implied damages of $4B [$300M-$45B]. The additional risk conditional on AI uplift was $13B [$500M-$110B] based on expert estimates, and $6B [$550M-$60B] based on superforecaster estimates. However, there was high disagreement across the respondents.
- **In the survey, there was no consensus on which risk management policy would be most effective at managing the risk of this threat model.** We surveyed experts on three approaches to risk mitigation: (1) open sourcing; (2) deployment safeguards (e.g. refusals), and (3) giving cyber defenders early access to models. There was no clear consensus on which approach would be more effective. More research on the merits of these different approaches seems warranted.

Overall, despite the significant uncertainty about the model, estimates and survey results, this report illustrates the potential progress that could be made on specific threat models via in-depth empirical research, expert review and surveys. This methodology could also be applied and expanded on by AI companies and AISIs to estimate the size of other AI risks.

Table ES1. Expected marginal costs of different levels of AI uplift over a year

Threat Actor Type:	Calculated social damages	
	Annual Baseline Damages: {Severity} * {Capability} * {Willingness} Best guess [95% range]	**AI elite exploit uplift scenario:** counterfactual expected damages from AI uplift, accounting for benefits to defenders. Best guess [95% range]
TA1: Individual hobbyist hacker	~$0	+ $18M [$1M-$200M]
TA2: Individual professional hacker	$375K [$10K-$10M]	+ $2B [$350M-$15B]
TA3: Team of 10 experienced hackers	$5M [$100K-$200M]	+ $430M [$85M-$2B]
TA4: Team of 100 state-level hackers	$170M [$10M-$3B]	+ $300M [$90M-$1B]
TA5: Team of 1,000 state-level hackers	$210M [$45M-$1B]	+ $3M [$2M-$5M]
Total	$390M [$55M-$4B]	+ $2.7B [$530M-$18B]

Technical Summary

1. Background

AI companies, governments and other experts have raised concern about the potential for the misuse of frontier AI systems for cyberattacks. Many leading AI companies have published Safety Frameworks which define *capability thresholds* of the form: *if an AI model has capability X, then risk mitigations, such as deployment and security safeguards, should be imposed*. As part of this, companies commit to test whether models are approaching these capability thresholds using model evaluations and monitoring real world usage of AI models.

In order to justify capability thresholds, we need both:

1. **Threat models** that quantify how much a given AI capability X increases social risk, and:
2. **Risk thresholds** which determine what level of risk is unacceptable.

However, at present, there is a lack of published threat modeling work quantifying how bad it would be for AI models to gain certain cyber capabilities. Consequently: (a) there is a lack of published justification for existing capability thresholds; (b) it is unclear how to interpret the results of cyber evals; and (c) it is unclear how companies should prioritise risk mitigations.

This report aims to help companies set capability thresholds by exploring one AI cyber threat model in-depth. Specifically, we focus on one mechanism by which AI *vulnerability and exploit discovery capabilities* might lead to significant social harm. This report aggregates a range of expert opinions to try to create a robust and consensus estimate for the following question:

> "***If*** future AI systems enable different threat actors to develop what we call "elite exploits", then how much would this increase the economic risk from data damaging cyber worm attacks, similar to WannaCry or NotPetya?".

Answering this question alone is not sufficient to justify a capability threshold. Threat modeling can tell us how risky a capability might be, but it does not tell us whether that risk is unacceptable. We leave the question of appropriate risk thresholds to future work.

2. Threat model scope

There are many possible AI-cyber threat models. This report reviews one specific narrow scenario in depth. This is not a comprehensive review of AI-cyber threat models, and future work should cover other scenarios.

Following NIST (2025), Appendix E, we can categorise threat models by: (1) the type of cyber attack; (2) the relevant capability that could enable such attacks; (3) the threat actor pursuing it;

and (4) the potential negative outcomes that might result. This report specifically focuses on (1) “data damaging worms” that spread autonomously and damage (encrypt, wipe or corrupt) data on a large number of infected systems, (2) enabled by what we call ‘elite’ exploits of vulnerabilities, (3) that any actor might use to (4) cause widespread economic damage.

Table TS1 summarises what is in and out of scope for this threat model.

Table TS1. *Different AI-cyber threat models. Orange is in scope for this report*

Type of Attack	Key Capability	Threat Actor	Outcome
Data damaging worms: malware that is designed to propagate automatically to infect, and encrypt or damage data on, a large number of systems (e.g. WannaCry and NotPetya). We focus on worms that cannot agentically change their code after release. **Worms to create botnets:** Worms can infect a large number of systems which can be used as ‘botnets’, a network of computers controlled by an attacker, to launch denial of service attacks or send spam (e.g. Storm Worm). **Worms which only create damage by increasing network traffic:** Some worms do not directly damage infected systems, but cause damage by creating a large amount of network traffic (e.g. Sasser). **Polymorphic worms** that can change agentically change their code after release (e.g. Conficker) **Critical infrastructure attacks:** cyber attacks on e.g. water, electricity, gas. (e.g. Russian attacks on Ukraine grid; Stuxnet worm attack on Iranian nuclear centrifuges). **Industrial espionage:** Cyber attacks on firms in order to steal IP. (e.g. China’s theft of IP for the F-35 fighter jet). **State espionage:** Cyber attacks by states in order to steal sensitive information (e.g. 2020 Solarwinds attack). [...]	**“Elite exploit” development.** This includes *both* the ability to find critical vulnerabilities in software and to write the code to exploit them. “Elite exploits”, as we define them, can infect a large number of systems, require no user interaction to spread, and offer a high degree of control over target systems. **Other steps in the cyber kill chain** including reconnaissance, phishing, social engineering, malware development, other post-intrusion tasks such lateral movement, privilege escalation, deployment of payloads, and data exfiltration, and analysis of exfiltrated data [...]	**All**	**Large economic damages (>$1B)**: costs to firms and consumers from economic disruption. This could include lost revenue, reduced consumption, remediation costs and costs to insurers. **Broader welfare costs:** costs on other determinants of human welfare, such as health. **Geopolitical costs:** Some attacks have important geopolitical implications (E.g. China’s 2015 hack of the Office of Personnel Management compromised security clearance data on millions of Americans.) [...]

We consider uplift to all levels of threat actors. We define threat actors in terms of the sophistication, or ‘operational capacity’ of the operations they can carry out. Table TS2 summarises these definitions.

We first define *operational capacity* levels. RAND (2024) p. 9-10, defines five levels of cyberattack operational capacity ('OC') in terms of the resources and capabilities available to the operation, ranging from OC1 operations to OC5 operations.

Following from this, we define five *threat actor* categories, ranging from TA1 to TA5, in terms of the highest OC operation they are able to carry out. For example, because some cybercrime groups are able to carry out OC3 operations but not higher, they are a TA3 threat actor; because the US is able to carry out OC5 operations, they are a TA5 threat actor, and so on. We treat states as single or unitary threat actors. For instance, we treat China as a single threat actor, rather than treating each Chinese state or state-backed cyber team as single threat actors.[3]

For ease of analysis, we assume that each threat actor in a threat actor class has the same capability level. So for example, China and the US are TA5 actors, so we assume they have the same capability level; North Korea and Iran are TA4 actors, and so we assume they have the same capability level. Table TS2 includes estimates of the number of threat actors in each class, based on a small (*n*= 21) survey of experts and superforecasters.

[3] In this respect, our definition is different to that commonly used in cyber threat modelling.

Table TS2. *Threat actor level definitions*

OC-level	Definition	Threat Actor Level	Definition	Example	Estimated number of actors
OC1	Operations roughly less capable than or comparable to a single individual with some limited professional expertise in information security spending several days with a total budget of up to $1K on the specific operation, and no preexisting infrastructure or access to the organization	**Threat Actor level 1 (TA1)**	Actors only capable of OC1 operations but not higher.	Hobbyist hackers	~1M
OC2	Operations roughly less capable than or comparable to a single individual who is broadly capable in information security spending several weeks with a total budget of up to $10K on the specific operation, with preexisting personal cyber infrastructure.	**Threat Actor level 2 (TA2)**	Actors only capable of OC2 operations but not higher.	Individual professional hackers	10K-100K
OC3	Operations roughly less capable than or comparable to ten individuals who are experienced professionals in information security spending several months with a total budget of up to $1 million on the specific operation, with major preexisting cyberattack infrastructure	**Threat Actor level 3 (TA3)**	Actors only capable of OC3 operations but not higher.	Well-known criminal hacker groups, well-resourced terrorist organizations, and industrial espionage organizations.	100-1K
OC4	Operations roughly less capable than or comparable to 100 individuals who have experience in a variety of relevant professions (cybersecurity, human intelligence gathering, physical operations, etc.) spending a year with a total budget of up to $10 million on the specific operation, with vast infrastructure and access to state resources such as legal cover, interception of communication infrastructure, and more.	**Threat Actor level 4 (TA4)**	Actors only capable of OC4 operations but not higher.	Some leading cyber states (e.g. North Korea, Iran)	10-50
OC5	Operations roughly less capable than or comparable to 1,000 individuals who have experience and expertise years ahead of the (public) state of the art in a variety of relevant professions (cybersecurity, human intelligence gathering, physical operations, etc.) spending years with a total budget of up to $1 billion on the specific operation, with state-level infrastructure and access developed over decades and access to state resources such as legal cover, interception of communication infrastructure, and more.	**Threat Actor level 5 (TA5)**	Actors capable of OC5 operations	The world's most capable states (e.g. the US, China, Russia)	~5

Key definitions

A **worm** is a malware program that can autonomously spread between computer systems. Worms can cause damage in various different ways. We focus on what we call “data damaging worms”, which we define as worms that directly damage (wipe, encrypt or corrupt) data on a large number of systems. However, not all worm attacks cause social harm in this way. Some worms cause damage simply by creating high network traffic, while others create ‘botnets’ to send spam or launch Denial of Service attacks against specific websites or companies.

A **vulnerability** is a bug in software or hardware that creates a security weakness in the design, implementation, or operation of a system or application that can in some way be exploited by an attacker. Vulnerabilities can be introduced intentionally or unintentionally through an accidental design or implementation flaw.

An **exploit** is malicious code that takes advantage of one or more software vulnerabilities to infect, disrupt, or take control of a computer without the user’s consent and typically without their knowledge. Finding vulnerabilities and developing exploits of those vulnerabilities are distinct tasks, and require different skillsets.

We define ‘**elite exploits**’ as exploits that are zero-click, allow remote code execution with high privileges and are effective against widely used software. Table TS3 explains these features.

Table TS3. *Defining elite exploits*

Exploit feature	Definition
Zero-click	Infection requires no user interaction, such as opening emails, clicking links or visiting a webpage.
Remote code execution	Allow attackers to execute arbitrary code on a system without the user’s knowledge, and without attackers requiring physical access to the system
High privileges	Privileges are the permissions granted to users, programs, or processes to perform specific actions on a system or access particular resources. Privileges levels range from (low to high): sandboxed application, user, administrator, to system level. Elite exploits have administrator privileges or higher.
Targets widely used software	Effective against >10M systems

A **patch** is a software update that fixes a vulnerability. Once vendors become aware of vulnerabilities, they will, after a delay, develop and release patches for them. After a further delay, these patches are then deployed or installed by users.

Why we prioritised this threat model

Prior to this report, we reviewed 34 case studies of past prominent cyberattacks, including state espionage, critical infrastructure attacks, worm attacks, and so on (Halstead and van der Merwe nd). In brief, we prioritised the data damaging worm threat model for further research for several reasons:

1. **Severe and sudden harm**: Data damaging worms could cause severe and sudden economic harm, making it difficult for society to adapt. Two data damaging worm attacks in 2017, WannaCry and NotPetya, were among the most economically damaging cyberattacks ever, causing ~$1B and ~$10B in damage respectively, in less than a day.
2. **Many actors are willing to launch worms**: Prior to 2009, significant worm attacks were common, but have since become much harder due to improved cybersecurity. Thus, many threat actors appear to be willing to launch worm attacks that cause widespread economic harm, but are currently unable to do so.
3. **Narrow cyber capability would increase risk**: If future AI systems gained one narrow capability - the ability to develop ‘elite exploits’ - this would significantly reduce the capability barriers to developing data damaging worms. By contrast, other types of cyberattack, such as espionage or critical infrastructure attacks, involve a suite of broader capabilities. They require a large number of distinct tasks, and a high degree of agentic or autonomous action and situational decision-making in novel or undocumented environments. Thus, uplift to these sorts of attacks sets a much higher capability bar, which may already be covered by AI autonomy evaluations.

Although we think the data damaging worm threat model is especially concerning, other AI-cyber misuse threat models may also be worth further research.

This threat model sets a high capability bar

Our primary aim is to analyse what the social costs would be *if* AI capabilities uplift threat actors to find elite exploits (including both finding vulnerabilities and writing the code to exploit them). We take no stance on if and when AI models might reach this capability level.

Vulnerability discovery and exploit development are already a major focus for AI cyber model evaluations. Model evaluations suggest that frontier models currently fall well-short of being able to find the elite exploits that are the focus of this report. However, model evaluations sometimes under-elicit models’ cyber capabilities, models are improving rapidly, and future capabilities are hard to predict. It is possible that models will gain the ability to find elite exploits in the near future, though we emphasise that this sets a high capability bar, one that is much higher than tested for in current model evaluations.

3. Methodology

Our aim in this report is to estimate the *marginal* risk posed by future hypothetical AI capabilities, compared to current *baseline* risk. We want, in other words, to estimate the effect that AI might have on risk, compared to the counterfactual in which AI capabilities do not improve further.

In order to justify potentially costly risk mitigations on the part of AI companies, this estimate should be well-supported by evidence. To that end, our approach is as follows.

1. Construction of risk model: We built a model that breaks down the final estimate into more tractable subquestions about the parameters that determine risk.
2. Empirical research: We conducted empirical research, which could inform our own initial estimates for the model parameters, and expert forecasts.
3. Expert feedback: We sought expert feedback from cyber and national security experts who provided comments on earlier drafts.
4. Expert survey: We surveyed 8 subject-matter experts and 13 credentialed 'superforecasters' on the parameter inputs into the model, and other questions relevant to the report. Given the small sample, the results of the survey should be interpreted with caution, but it provides initial information on a wider range of perspectives on the risk. Future work could expand the survey to produce more robust consensus estimates. We will discuss the survey methodology and results in depth in follow-up work.

For our simple risk model, we broke down the estimate into three parameters.

- The ***capabilities*** of different threat actors: For each threat actor class, the probability over the next year that a *randomly selected* threat actor in that class can launch data damaging worms if they wanted to, assuming no further improvements in AI. Note that we assume for ease of analysis that all threat actors in the same class have the same capability level.
- How ***willing*** threat actors are to launch data damaging worm attacks: the probability over the next year that *at least one* threat actor in each class would be willing to launch a data damaging worm attack if they were able
- The ***damages*** from such attacks: how much economic harm such attacks would do.

We focus only on expected damages from data damaging worms that could, on their own, cause >$1B in damage. So, we do not try to estimate the cumulative expected damages from all data damaging worms, including smaller attacks. Firstly, this simplifies the analysis. Secondly, because worms spread exponentially, we would expect the distribution of damages from worms to be heavy tailed, so that most of the expected damages are from these tail events.

We can bring estimates of the three parameters together to calculate baseline and marginal expected damages from data damaging worm attacks launched by different threat actors.

The formula for baseline expected damages is shown below:

***Baseline* expected damage from threat actor**$_i$ = Capability$_i$ * Willingness$_i$ * Damages

The formula for marginal expected damages from AI uplift is:

> ***Marginal* expected damage assuming maximal AI elite exploit uplift to threat actor$_i$**
> = ((Capability$_i$ | AI uplifts threat actor$_i$ to find elite exploits) * Willingness$_i$ * Damages * Adjustment for the effect of AI on defence) - Baseline damages$_i$

Several things should be noted about this formula.

1. **AI uplift** on attacker capability is captured by conditioning the probability that attackers can launch data damaging worms on AI uplifting threat actors to find elite exploits.
2. **The marginal impact of AI i**s captured by subtracting baseline damages, which would have occurred even without further improvements in AI.
3. The formula adjusts for the effects of **AI benefits to cyberdefenders.** The AI capabilities we investigate here - vulnerability discovery and exploit development - are *dual use* in that they are useful to both attackers and defenders. It is important to consider how benefits to attack and defence net out.

4. Analysing the determinants of data damaging worm risk

We will now discuss threat actor capability, threat actor willingness, and potential damages in more depth.

Many actors would be willing to launch harmful worm attacks

There is strong evidence that many actors would be willing to launch worms that could cause a large amount of economic harm if they were able to do so. Figure TS1 below shows significant worm attacks by different threat actors, using data adapted from Johansmeyer (2024). Note that only a minority of these attacks were data damaging worms: many of the worms did not directly damage data on infected systems but instead caused damage by creating high network traffic or by creating botnets to launch Denial of Service attacks or send spam. Nonetheless, we think this still provides useful information about actor willingness to launch data damaging worms because data damaging worms seem in many respects better suited to their aims

Figure TS1. *Major worm attacks by threat actor (1998-2023) [Adapted from Johansmeyer, 2024]*

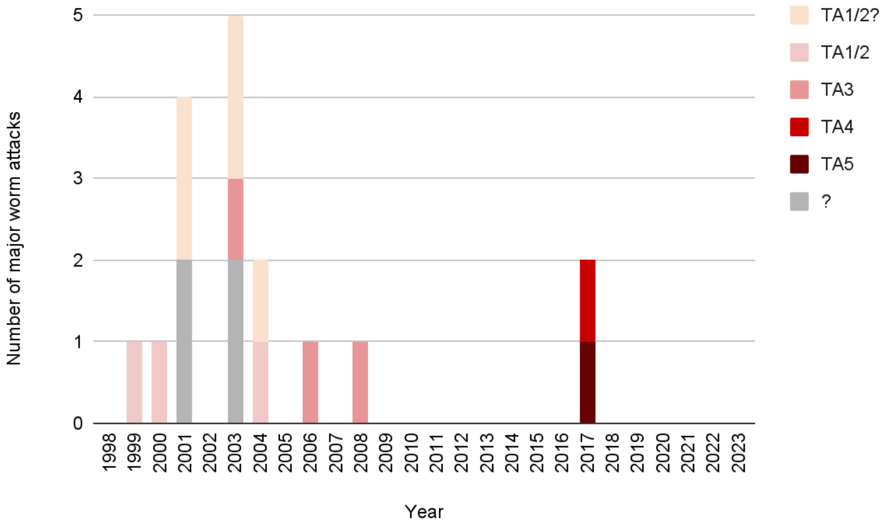


Figure TS1 shows that:

1. The frequency of significant worm attacks has declined over time, with a large number of attacks prior to 2005, and far fewer since then.
2. In recent years, more sophistication has been required: Prior to 2005, significant worm attacks were mostly perpetrated by OC1/2 (individual hackers); from 2005-2008 there were several OC3 attacks (a group of ~10 experienced hackers); and since then the only attacks were in 2017 by OC4 and OC5 actors (nation-states).

These trends were likely driven by improvements in cybersecurity, such as improvements in email filtering, firewalls and vulnerability patching. This suggests that: (1) some actors would be willing to launch worm attacks that cause mass economic harm; but (2) lower skilled actors now lack the ability to do so, due to improved cybersecurity.

AI elite exploit development capabilities would significantly reduce barriers to data damaging worms

We argue that **if** AI gained strong elite exploit development capabilities (including both finding vulnerabilities and developing exploits of them), this would overcome the capability barriers many threat actors face to developing data damaging worms. Since many threat actors would be willing to release such worms, this would significantly increase the risk of worm attacks.

Elite exploits are particularly powerful cyberweapons, and are especially useful for data damaging worm attacks. Worms using elite exploits can infect a large number of systems without user interaction and so have great potential reach. Since they allow remote code execution with high privileges, they can deploy further malware ‘payloads’, such as ransomware that encrypts data or data wipers, to damage data on infected devices.

The 2017 WannaCry and NotPetya attacks provide further evidence that elite exploits could increase the risk of data damaging worms. These attacks were the product of an unusual natural experiment in which an elite exploit known as 'EternalBlue', initially developed by the NSA, was leaked to the public in April 2017. North Korea and Russia then used EternalBlue in the WannaCry and NotPetya worms 1-2 months later.

The evidence suggests that developing the EternalBlue elite exploit was much harder than the other steps involved in creating the WannaCry worm:[4] developing EternalBlue likely took ~90% of the overall skilled researcher time involved in making the worm. A range of evidence suggests that today it is very difficult to create elite exploits. It likely costs on the order of hundreds of thousands to millions of dollars to create elite exploits, and they are hard to develop even for some state intelligence agencies.

Nonetheless, completing the other tasks involved in making the worm, aside from developing elite exploits, still seems non-trivial. WannaCry and NotPetya were both released by state actors, which suggests that even given access to an elite exploit like EternalBlue, it was difficult for non-state actors to develop a worm using it before most systems were patched for the vulnerabilities exploited by EternalBlue.

Past attacks had the potential to cause very large damage, but improved cybersecurity may now reduce risk

Data on the damages of past worm attacks is poor, but there is reasonable evidence that WannaCry and NotPetya caused damages of ~$1B and ~$10B, respectively. Moreover, with relatively modest changes in their code, they could have caused tens of billions to $100B in damage. The WannaCry code contained an error that prevented further spread and damage to systems, while NotPetya was designed to limit damage mainly to Ukraine.

It is less clear whether data damaging worms using elite exploits would cause comparable damage if they were released *today*. The reason for this is that modern cybersecurity might make it substantially harder for worms to spread and cause damage on infected systems.

5. Offence-defence balance

AI models that discover vulnerabilities and develop exploits are *dual use* in that they could benefit both attackers and defenders. It is difficult to assess the effects of AI on the risk of worm attacks over time because offence-defence balance depends on a range of uncertain parameters, and AI could benefit attackers in some respects and benefits defenders in others. Table TS4 outlines how AI relates to different determinants of the risk of worm attacks over time.

[4] NotPetya was different because Russia aimed to limit damage to Ukraine and so released the worm by compromising tax software widely used in Ukraine. So, the Russian attackers probably spent more time on this part of the attack. These steps would not be necessary for threat actors aiming to cause broader damage, which are our main focus here.

Table TS4. How AI might affect different determinants of the risk of worm attacks

Determinant of risk of worms	**Favours offence of defence?**	**Discussion**
Vulnerability discovery	**Unclear**	AI capability can be used by attackers and defenders.
Correlation of vulnerabilities found by attackers and defenders	**Unclear**	If vulnerabilities found by attackers and defenders are correlated, this favours defence. There are some reasons to think that automation will lead to greater correlation, but correlation may be lower due to experimentation in prompting and scaffolding and use of different AI models.
Exploit development	**Favours offence**	This allows attackers to exploit faster. AI exploit development capability may be correlated with general AI coding abilities.
Patch development	**Favours defence**	Allows defenders to patch faster. This capability may also be correlated with general AI coding abilities. So, exploit development and patch development capabilities may be correlated.
Patch deployment	**Favours offence in short-term (<3m)**	Currently a key lag on defence. It seems harder for AI to affect patch *deployment* than e.g. exploit or patch development. Policy decisions have a greater impact on patch deployment rates.
Defenders can repair vulnerabilities prior to software release	**Favours defence in longer-term (>6m)**	Defenders have a natural advantage in one respect in that they can use AI to repair vulnerabilities in their software *prior to the release of the software*. As new software replaces old software, the risk of cyberattacks declines. Apple and Google release a new OS around once a year, while releases one major feature update per year, and a major new OS every three years. So, plausibly defenders would start to benefit after around 6-12 months after the release of AI tools, as new AI-tested software replaces old software.
Warning shots	**Favours defence in longer-term (>3m)**	Major cyberattacks serve as a warning shot which encourages defenders to improve cybersecurity. This broad dynamic seems to have occurred for worm attacks, which were high prior to 2005 but then declined once cybersecurity improved after numerous major worm attacks (see Section 1).

Table TS4 suggests that the effect of AI on the risk of worm attacks will likely vary over time. It seems plausible that AI vulnerability discovery and exploit development capabilities will increase the risk of worms in the short-term because AI will improve (a) *vulnerability discovery* for both attackers and defenders, (b) *exploit development* for attackers, and (c) *patch development* for defenders, but there is less scope for AI to improve *patch deployment* rates for defenders. This suggests that AI uplift to factors (a), (b) and (c) will increase risk in the gap in which users deploy patches.

Available data suggests that for widely used software today, 90% of patches would be deployed 0.5-3 months after patch release. AI-enabled elite exploit development would increase the risk of worm attacks in the time it takes for defenders to deploy patches. The longer-term effect (over 3-12 months) is less clear.

6. Results

The author's estimates are summarised in Table TS5. The table shows the risk posed by five different classes of threat actors, ranging from TA1 (individual hobbyist hackers) to TA5 (the world's most capable states).

- **Baseline damages**: The author estimated baseline damages (i.e. assuming no further progress in AI) from data damaging worm attacks of around $100M. Expert and forecaster estimates imply that the risk is around $1B.
- **Strong improvements in AI elite exploit discovery capabilities would increase risk.** To operationalise the effect of future improvements in AI capabilities, we hypothesised strong improvements in relevant AI capabilities: a future study finds that a future AI system enabled 25% of TA2 actors (individual professional hackers) to develop elite exploits, with three months of full time effort. The author estimates that *if* this were to happen, *expected* social damages (the social damages that could occur, weighted by their probability) would increase by ~10X to around $1B. Superforecaster and expert estimates implied a smaller increase in relative risk (3-4X), but from a higher baseline, implying overall social damages of around $3B-$8B. All of these estimates assume that the model is open weight, and would also benefit defenders by helping them find and patch vulnerabilities.

Table TS5. Expected marginal costs of different levels of AI uplift over a year

Threat Actor Type:	**Inputs**				**Calculated social damages**	
	Capability		**Willingness:** Annual probability at least one actor in each class launches an attack	**Severity:** Damages from a data damaging worm attack	**Annual Baseline Damages:** {Severity} * {Capability} * {Willingness} Best guess [95% range]	**AI elite exploit uplift scenario:** counterfactual expected damages from AI uplift, accounting for benefits to defenders. Best guess [95% range]
	Exploit capability: Probability a random actor in each class can develop elite exploits	**Non-exploit worm capability:** probability a random actor in each class can write a worm if they already have elite exploits				
TA1: Individual hobbyist hacker	Very remote chance (~0%) *Significant uplift needed*	Extremely unlikely (0.1%-2%) *Significant uplift needed*	Extremely likely (~100%) *Many actors, one will do*	$10B–$100B	~$0	+ $18M [$1M-$200M]
TA2: Individual professional hacker	Very remote chance (~0%) *Significant uplift needed*	Unlikely (1–20%) *Significant uplift needed*	Highly likely (70-100%) *Many actors, one will do*		$375K [$10K-$10M]	+ $2B [$350M-$15B]
TA3: Team of 10 experienced hackers	Very unlikely (1-10%) *Significant uplift needed*	Realistic possibility (5-60%) *Some uplift needed*	Highly unlikely (1–5%) *Small subset may want to*		$5M [$100K-$200M]	+ $430M [$85M-$2B]
TA4: Team of 100 state-level hackers	Realistic possibility (5-60%) *Some uplift needed*	Almost certain (~100%) *No uplift needed*	Highly unlikely (1–5%) *Small subset may want to*		$170M [$10M-$3B]	+ $300M [$90M-$1B]
TA5: Team of 1,000 state-level hackers	Almost certain (90%–100%) *No uplift needed*	Almost certain (~100%) *No uplift needed*	Very remote chance (0.5%-1%) *No strategic incentive*		$210M [$45M-$1B]	+ $3M [$2M-$5M]
Total					$390M [$55M-$4B]	+ $2.7B [$530M-$18B]

Our main focus in this report is on estimating the social costs of AI capabilities. However, we also briefly consider the merits of different risk mitigation policies AI companies might take. The estimates above assume that models are open weight and have no deployment safeguards (e.g. refusals and jailbreak protections). Two alternative policies to open weighting are:

> **P1: Proprietary models with refusals and anti-jailbreak measures.** Frontier AI models are proprietary and require users to access them via APIs with deployment and security safeguards. Companies train the models to refuse to respond to requests to find vulnerabilities or develop exploits, and have protections against jailbreaks. The models are assumed to be protected by SL-2 level infosecurity, which can likely thwart many moderate-effort attacks by individual hackers (RAND (2024)).
>
> **P2: Temporary protections with early access for defenders.** The public release of the model has P1 level safeguards in place.[5] However, a specific set of 'cyber defenders', including leading technology companies and bug bounty programs, is given access to a version of the model without refusals, i.e. with full vulnerability discovery and exploit development capabilities. After four months, the model is released under P0 security, i.e., is open weight.[6]

These policies each have different advantages and disadvantages. P0 – open weight models with no refusals – allow *both* attackers and defenders to benefit in full from AI vulnerability discovery and exploit development capabilities. P1 – proprietary models with refusals and anti-jailbreak measures – makes it harder for *both* attackers and defenders to benefit from these capabilities. Finally, P2 – temporary protections with early access for defenders – aims, over a short period, to limit benefits to attackers and provide benefits to defenders.

Since it is unclear whether AI vulnerability and exploit capabilities favour offence or not, it is also unclear which of these policies would be optimal. In our small pilot survey, we surveyed experts on the effect these three policies would have on the risk of data damaging worms. As Figure TS2 shows, there was no clear consensus on the effectiveness of the different policies, and there was a disagreement within and between experts and superforecasters on how the policies would affect risk.

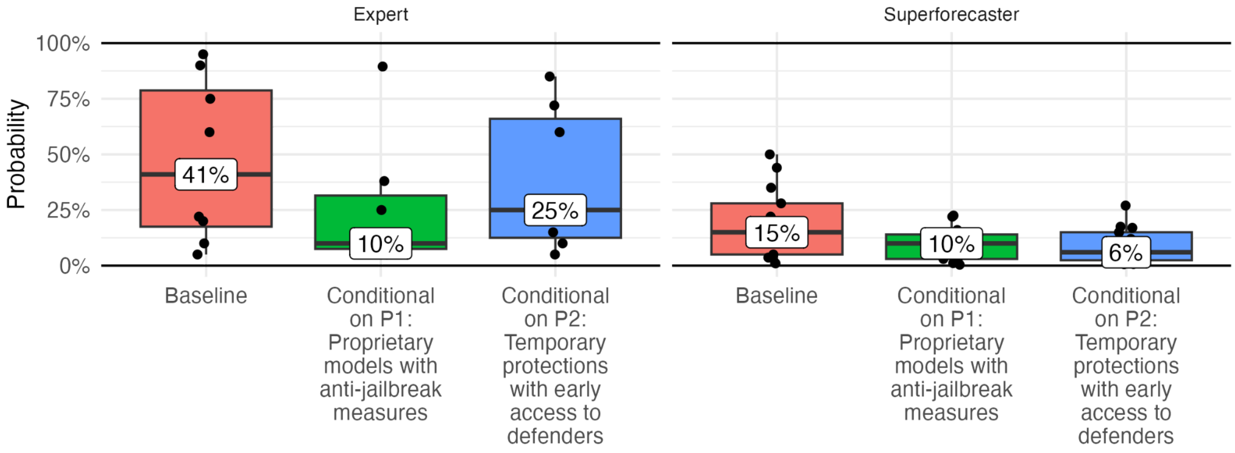
Figure TS2. Survey results on the effect of different risk management policies on the risk of data damaging worms, conditional on AI elite exploit uplift.

[5] For discussion of a similar idea, see Ee et al 'Asymmetry by Design: Boosting Cyber Defenders with Differential Access to AI' (2025).

[6] P0 and P2 could be combined with subsidies for vulnerability discovery. More research is needed on what level of funding would be optimal, but for context, Apple, Google and Microsoft collectively pay on the order of tens of millions of dollars each for bug bounties (Google 2025; Apple 2022, Microsoft 2024).

Probability of at least one data-damaging worm attack similar to WannaCry or NotPetya causing at least $10 billion in economic damages in 2026

All scenarios assume Capability 1 has been met



One additional benefit of allowing defenders to use AI capabilities, as in policies P0 and P2, is that they might help to gather better information about model capabilities. There is evidence that model evaluations under-elicit true model capabilities, as they fail to capture how creatively humans might use models in the real world. Giving defenders early access to models or open sourcing models, and incentivising vulnerability discovery, might provide better information about model capabilities than traditional task-based evaluations.

Overall, more research on the merits of these different approaches may be warranted.

Main report

1. Introduction

1.1. Background

AI companies, governments and other experts have raised concern about the potential for the misuse of frontier AI systems for cyberattacks (OpenAI 2025; Meta 2025; Anthropic 2025; Google DeepMind 2025; International AI Safety Report, 2025, p. 72ff). Many leading AI companies have published safety frameworks which include "if-then commitments" of the form: *if an AI model has capability X, risk mitigations Y must be imposed* (METR, 2024; Karnofsky, 2024). Thus, these if-then commitments set out *capability thresholds* that prescribe certain risk mitigations, such as improved deployment and security safeguards. As part of this, companies commit to test whether models are approaching these capability thresholds using model evaluations and also monitoring real world usage of AI models (Rodriguez et al 2025; Anthropic 2025; Bhatt et al 2024; OpenAI 2025).

But an open question remains about where exactly to set capability thresholds in the first place. In order to set appropriate capability thresholds, we need both (Koessler et al 2024):

1. **Threat models** that quantify how much a given AI capability X increases social risk, and
2. **Risk thresholds** which determine what level of risk is unacceptable.

However, at present, there is a lack of published threat modeling work explaining how AI capabilities translate into increased social risk. Thus it is difficult to know if existing capability thresholds in AI Safety Frameworks are set at an appropriate level and how to interpret the results of AI cyber model evaluations (Lukošiūtė and Swanda 2025). It is unclear whether weak performance on an evaluation implies that a model is safe, and conversely whether strong performance on an evaluation implies that a model is too risky.

For the same reason, it is unclear how AI companies should prioritise risk mitigations for AI models. Mitigations can involve potentially substantial costs, and the benefits of imposing mitigations ought to be commensurate with these costs. In the absence of estimates of the benefits of mitigations (in terms of reduced risk), AI companies cannot effectively prioritise risk mitigations.

This report aims to help companies set capability thresholds by answering question (1) and exploring one AI cyber threat model in-depth. Specifically, we focus on one mechanism by which AI vulnerability and exploit discovery capabilities might lead to significant social harm. Vulnerability and exploit discovery is already a major focus for AI companies' Safety

Frameworks and for model evaluations, but there is a lack of public threat modeling work explaining how harmful such capabilities would be.

For this report, we try to create a robust forecast for one specific and narrow question:

> "***If*** future AI systems enable different threat actors to develop what we call "elite exploits", ***then*** how much would this increase the economic risk from data damaging cyber worm attacks, similar to WannaCry or NotPetya?".

As noted, answering this question alone is not sufficient to justify a capability threshold. Threat modeling can tell us how risky a capability might be, but it does not tell us what level of risk is unacceptable and ought to be mitigated by AI companies. We leave this latter question to future work.

Importantly, our goal in this report is to answer a conditional question: '*if* AI models gain certain capabilities, what are the social costs?'. We take no stand on whether AI systems are *likely* to gain such capabilities in the future.

1.2. Key Definitions

Here we define key terms used in this report, including worms, vulnerabilities, exploits, zero-days, n-days, patches and payloads.

Worms

A worm is a malware program that can autonomously spread between computer systems without the need for an *attacker* to take active steps to infect each system. Some worms require *victims* to take certain actions to spread. For example, many early worms spread via email and required users to open malicious attachments in emails, which enabled the malware to send the worm to other contacts in the victim's address book. Other worms require no interaction on the part of victims to spread between systems.

Worms can cause damage in various different ways. We focus on what we call "data damaging worms", which we define as worms that directly damage (wipe, encrypt or corrupt) data on a large number of systems. However, not all worm attacks cause social harm in this way. We discuss this in section 1.3.

Vulnerabilities, exploits and patches

A **vulnerability** is a bug in software or hardware that creates a security weakness in the design, implementation, or operation of a system or application that can in some way be exploited by an attacker (RAND 2017, p. 2). Vulnerabilities can be introduced intentionally or unintentionally through an accidental design or implementation flaw.

An **exploit** is malicious code that takes advantage of one or more software vulnerabilities to infect, disrupt, or take control of a computer without the user's consent and typically without their knowledge (RAND 2017, p. 2). Finding vulnerabilities and developing exploits of those vulnerabilities are distinct tasks, and require different skillsets (PatternLabs 2025).[7]

Exploits are often categorised in terms of what they allow attackers to do (RAND 2023, p. 4). For example:

- **Local privilege escalation** exploits allow an attacker with limited access to a system to gain higher privileges, such as administrative or root access. These require prior access to the system, often through a lower privileged account (Admin Network & Security nd).
- **Sandbox escape** exploits allows attackers to break out of a restricted execution environment (sandbox) to access sensitive data or execute code on the host system (NordVPN nd).
- **Remote code execution** exploits allow attackers to execute arbitrary code on a system without the user's knowledge, and without attackers requiring physical access to the system (Crowdstrike 2022).

Attackers often use chains of exploits together in order to give attackers the level of access and system privileges required to achieve desired effects (RAND 2023, p. 4).[8]

In this report, we focus on one class of very powerful exploits, which we call "elite exploits", which we define as zero-click exploits, that allow remote code execution with high privileges, and that are effective against widely used software. Table 1.1 explains these features in more detail.

Table 1.1. *Defining elite exploits*

Exploit feature	**Definition**
Zero-click	Infection requires no user interaction, such as opening emails, clicking links or visiting a webpage.[9]
Remote code execution	Allow attackers to execute arbitrary code on a system without the user's knowledge, and without attackers requiring physical access to the system
High privileges	Privileges are the permissions granted to users, programs, or processes to perform specific actions on a system or access particular resources. Privileges levels range from (low to high): sandboxed application, user, administrator, to system level. Elite exploits have administrator privileges or higher.

[7] See also Charlie Miller, 'How to build a cyber army to hack the US', Defcon (2013).
[8] See Appendix A.5.
[9] Note that on some definitions of 'zero-click', not all zero-click exploits require absolutely no user interaction to spread. For example, some watering hole attacks can infect a system if a user visits a website but does not click on any links on the website. Some definitions class this as a zero-click exploit (e.g. SmarterMSP 2024). On our definition, these would not be elite exploits because they require the user to visit a compromised website, and therefore involve user interaction.

Targets widely used software	Effective against >10M systems

Zero-day vulnerabilities (or “zero-days”) are vulnerabilities that are unknown to the software vendor (Smeets p21), while *n-day vulnerabilities* are vulnerabilities that are known to the vendor (Smeets, p. 21). *Zero-day exploits* are exploits of zero-day vulnerabilities, while *n-day exploits* are exploits of n-day vulnerabilities.

A **patch** is a software update that fixes a vulnerability. By definition, there are no patches for zero-day exploits. However, once vendors become aware of vulnerabilities, they will, after a delay, develop and release patches for them. After a further delay, these patches are then deployed or installed by users. We discuss data on patch development and deployment times in section 5.

Payloads

The **payload** is the element of the cyberweapon that achieves the ultimate desired effect on the infected system (RAND 2023, p. 5). For instance, in a ransomware attack, the payload is the malware that encrypts the victim's files. For spyware, the payload would be the component of the malware that captures and transmits sensitive information from the victim's device to the attacker.

1.3. Scope of the report

There are many possible AI-cyber threat models. This report reviews one specific narrow scenario in depth. This is not a comprehensive review of AI-cyber threat models, and future work should cover other scenarios.

Following NIST (2025), Appendix E, we can categorise threat models by: (1) the type of cyber attack; (2) the relevant capability that could enable such attacks; (3) the threat actor pursuing it; and (4) the potential negative outcomes that might result. This report specifically focuses on (1) “data damaging worms” that damage (encrypt, wipe or corrupt) data on a large number of infected systems, (2) enabled by what we call ‘elite’ exploits, (3) that any actor might use to (4) cause widespread economic damage.

Table 1.2 summarises what is in and out of scope for this threat model.

Table 1.2. *Different AI-cyber threat models.*

	Type of Attack	Key Capability	Threat Actor	Outcome
In scope	**Data damaging worms:** malware that is designed to propagate automatically to infect, and encrypt or damage data on, a large number of systems (e.g. WannaCry and NotPetya). We focus on worms that cannot agentically change their code after release.	**“Elite exploit” development.** This includes *both* the ability to find critical vulnerabilities in software and to write the code to exploit them. “Elite exploits”, as we define them, can infect a large number of systems, require no user interaction to spread, and offer a high degree of control over target systems.	**All** (See Table 1.2)	**Large economic damages (>$1B)**: costs to firms and consumers from economic disruption. This could include lost revenue, reduced consumption, remediation costs and costs to insurers.
Out of scope (selected examples)	**Worms to create botnets:** Worms can infect a large number of systems which can be used as ‘botnets’ to launch denial of service attacks or send spam (e.g. Storm Worm). **Worms which only create damage by increasing network traffic:** Some worms do not directly damage infected systems, but cause damage by creating a large amount of network traffic (e.g. Sasser). **Polymorphic worms** that can change agentically change their code after release (e.g. Conficker) **Critical infrastructure attacks:** cyber attacks on e.g. water, electricity, gas. (e.g. Russian attacks on Ukraine grid; Stuxnet worm attack on Iranian nuclear centrifuges). **Industrial espionage:** Cyber attacks on firms in order to steal IP. (e.g. China’s theft of IP for the F-35 fighter jet). **State espionage:** Cyber attacks by states in order to steal sensitive information (e.g. 2020 Solarwinds attack). [...]	**Other steps in the cyber kill chain** including reconnaissance, phishing, social engineering, malware development, other post-intrusion tasks such lateral movement, privilege escalation, deployment of payloads, and data exfiltration, and analysis of exfiltrated data [...]		**Broader welfare costs:** costs on other determinants of human welfare, such as health. **Geopolitical costs:** Some attacks have important geopolitical implications (E.g. China’s 2015 hack of the Office of Personnel Management compromised security clearance data on millions of Americans.) [...]

Types of Actors: Differentiating by Operational Capacity

We consider uplift to all levels of threat actors. We define threat actors in terms of the sophistication, or ‘operational capacity’ of the operations they can carry out. Table 1.3 summarises these definitions.

We first define *operational capacity* levels. RAND (2024) p. 9-10, defines five levels of cyberattack operational capacity (‘OC’) in terms of the resources and capabilities available to the operation, ranging from OC1 operations to OC5 operations. By definition, each category includes the capacities of all preceding ones. For example, the most competent nation-states, such as the US and China, are able to carry out OC5 operations, but also all operations below that level.

Following from this, we define five *threat actor* categories, ranging from TA1 to TA5, in terms of the highest OC operation they are able to carry out. For example, because some cybercrime groups are able to carry out OC3 operations but not higher, they are a TA3 threat actor; because China is able to carry out OC5 operations, they are a TA5 threat actor, and so on. We treat states as single or unitary threat actors. For instance, we treat China as a single threat actor, rather than treating each Chinese state or state-backed cyber team as single threat actors.[10]

For ease of analysis, we assume that each threat actor in a threat actor class has the same capability level. So for example, China and the US are TA5 actors, so we assume they have the same capability level; North Korea and Iran are TA4 actors, and so we assume they have the same capability level.

[10] In this respect, our definition is different to that commonly used in cyber threat modelling.

Table 1.3. *Threat actor level definitions*

Operational Capacity level	**Definition**	**Threat Actor Level**	**Definition**	**Example**	**Estimated number of actors**[11]
OC1	Operations roughly less capable than or comparable to a single individual with some limited professional expertise in information security spending several days with a total budget of up to $1K on the specific operation, and no preexisting infrastructure or access to the organization	**Threat Actor level 1 (TA1)**	Actors only capable of OC1 operations but not higher.	Hobbyist hackers	~1M
OC2	Operations roughly less capable than or comparable to a single individual who is broadly capable in information security spending several weeks with a total budget of up to $10K on the specific operation, with preexisting personal cyber infrastructure.	**Threat Actor level 2 (TA2)**	Actors only capable of OC2 operations but not higher.	Individual professional hackers	10K-100K
OC3	Operations roughly less capable than or comparable to ten individuals who are experienced professionals in information security spending several months with a total budget of up to $1 million on the specific operation, with major preexisting cyberattack infrastructure	**Threat Actor level 3 (TA3)**	Actors only capable of OC3 operations but not higher.	Well-known criminal hacker groups, well-resourced terrorist organizations, and industrial espionage organizations.	100-1K
OC4	Operations roughly less capable than or comparable to 100 individuals who have experience in a variety of relevant professions (cybersecurity, human intelligence gathering, physical operations, etc.) spending a year with a total budget of up to $10 million on the specific operation, with vast infrastructure and access to state resources such as legal cover, interception of communication infrastructure, and more.	**Threat Actor level 4 (TA4)**	Actors only capable of OC4 operations but not higher.	Some leading cyber states (e.g. North Korea, Iran)	10-50
OC5	Operations roughly less capable than or comparable to 1,000 individuals who have experience and expertise years ahead of the (public) state of the art in a variety of relevant professions (cybersecurity, human intelligence gathering, physical operations, etc.) spending years with a total budget of up to $1 billion on the specific operation, with state-level infrastructure and access developed over decades and access to state resources such as legal cover, interception of communication infrastructure, and more.	**Threat Actor level 5 (TA5)**	Actors capable of OC5 operations	The world's most capable states (e.g. the US, China, Russia)	~5

[11] These estimates are based on the results of our small survey and are therefore somewhat uncertain.

We will now clarify the scope of this threat model and explain why we prioritised it for further investigation.

Type of Attack: Data damaging worms (a subset of computer worm attacks)

Data damaging worms - worms that directly damage data on infected systems - are just one subset of worm attacks. Other types of worms have different functionality:

- **Damage via high network traffic:** Some worms cause damage only via propagating rapidly and thereby causing a large amount of network traffic, in turn causing networks to shut down. For example, the 1998 Melissa worm caused damage in this way, without directly damaging infected systems.
- **Botnets:** Some worms infect a large number of systems to create 'botnets' (a network of computers controlled by an attacker) to send spam or launch targeted Denial of Service attacks against specific websites. For example, StormWorm did not damage infected systems directly, but caused damage via creating a botnet to launch Denial of Service attacks antispam websites and security vendors.
- **Physical damage:** Worms can be used to cause damage to physical systems, or 'operational technology' (the hardware and software that directly manages and controls physical devices and processes in industrial settings). For example, the Stuxnet worm launched by the US and Israel physically damaged centrifuges at an Iranian nuclear centrifuge facility.
- **Espionage:** Some worms can be used for espionage. For example, Flame malware had worm-like properties, and was used for espionage in the Middle East (Securelist 2012).

Only a minority of past prominent worm attacks have been data damaging worms. Johansmeyer (2024) created a dataset of 'significant' cyberattacks. Johansmeyer's inclusion criteria are that: the worm must have affected what Johansmeyer calls a 'significant' number of companies (this is not precisely defined and involves some discretion, but >10 victim companies would plausibly count as significant, and >25 definitely would),[12] and there must be some public source claiming that it caused damages of >$800M. 17 of the 21 attacks in Johansmeyer's dataset were worm attacks.[13] Table 1.4 describes the worm attacks in the Johansmeyer dataset. 4 of the 17 significant worm attacks in this dataset were data damaging worms.[14]

[12] Tom Johansmeyer, personal communication, 16th Jan 2025.

[13] See the 'Johansmeyer 2024' tab of Worm damages.

[14] Note that Johansmeyer (2024) includes the 2010 Stuxnet attack, citing damages of $2.9B, though I have been unable to find the source for this estimate. See the 'Johansmeyer (2024)' tab of Worm damages. We exclude Stuxnet because we do not think it meets Johansmeyer's '>$800M damage' inclusion criterion. We discuss the damages of Stuxnet in the next subsection.

Table 1.4. *Details on past major worm attacks from Johansmeyer (2024)*[15]

Attack name	Year	Actor	Description	# systems Infected	Elite exploit?	Data damaging worm?
Melissa	1999	TA1/2	Email worm. Damage via high network traffic, no other destructive payload	~100K	n	n
ILOVEYOU	2000	TA1/2	Email worm. Corrupted documents, later variant wiped hard drive.	~50M	n	y
Klez	2001	TA1/2?	Email worm. Damage via high network traffic and disabling antivirus	~7M	n	n
CodeRed	2001	?	Zero-click. Defaced specific websites and launched targeted Denial of Service attacks against sites	~360K	y	n
Nimda	2001	?	Zero-click + email spread. Damage via high network traffic + allowed elevated privileges	>1.3M	y	n
SirCam	2001	TA1/2?	Email worm. Could expose confidential info, and delete all files in certain conditions	~2.3M	n	y
SoBig	2003	TA1/2?	Email worm to distribute spam. Also caused damage by creating high network traffic	>1M	n	n
SQL Slammer	2003	TA1/2?	Zero-click, but limited reach. Damage via high network traffic, no malicious payload	>75K	n	n
Swen	2003	?	Email worm. Terminated antivirus and firewalls, no other destructive payload	~1.5M	n	n
Mimail	2003	?	Email worm. Launched targeted Denial of Service attacks against anti-spam sites	~21K	n	n
Yaha	2003	TA3	Email worm. Can terminate security process and launch Denial of Service attacks against sites	?	n	n
MyDoom	2004	TA1/2?	Email worm. Damage via targeted DoS attacks and creating high network traffic	~500K	n	n
Sasser	2004	TA1/2	Zero-click. Damage only via high network traffic, no destructive payload.	~500K-1M	y	n
StormWorm	2007	TA3	Spread via email and social engineering. Created Denial of Service botnet to attack antispam websites and security vendors	1M-50M	n	n
Conficker	2008	TA3	Zero-click. Created large botnet, but never used for significant attack	~10M	y	n
WannaCry	2017	TA4	Zero-click. Encrypted files	~230K	y	y
NotPetya	2017	TA5	Zero-click. Encrypted files	~670K	y	y

[15] For sources, see Appendix A.8.

In what follows, it is important to bear in mind that we only focus on one subset of worm attacks.

Data damaging worms pose especially large economic risks

We prioritised data damaging worms for further investigation because they seem to pose large economic risks compared to other types of cyberattack.

WannaCry and NotPetya caused damages of ~$1B and ~$10B respectively, and, with modest changes to their code, they could have done much greater damage (see section 4). We think it is plausible that NotPetya was the most economically damaging cyberattack ever, and was very likely *one of* the most damaging cyberattacks ever.

For comparison, CISA (2020, sec. 3) reviewed damage estimates for large cyberattacks, and found that the most economically costly incidents were hacks of the US Office of Personnel Management ($760M) and the health insurer Anthem ($376M). Further support is provided by our own case study work. We reviewed 34 case studies of past prominent cyberattacks, including state espionage, critical infrastructure attacks, worm attacks, and so on (Halstead and van der Merwe nd). Of all of these case studies, NotPetya plausibly caused the greatest economic damage.

The Johansmeyer dataset mentioned above suggests that some worm attacks prior to 2010 caused very large economic damages. For example, Johansmeyer's dataset suggests that SoBig and MyDoom caused damages of $65B and $67B, respectively. These and most of the other worms in the Johansmeyer dataset were not data damaging worms, but instead caused damage via creating high network traffic or by creating large botnets. However, we put little weight on these damage estimates, and think they are substantial overestimates. As Johansmeyer notes, the provenance of this data is poor: most of the damage estimates in the dataset are impossible to verify and include no methodological information (Johansmeyer nd).[16] Thus, the most robust evidence suggests that data damaging worms have historically been the most damaging subset of worm attacks.

There are also other reasons to think that data damaging worms have greater potential to cause economic damage than other types of worms.

- **Network disruption**: Worms that cause damage only via creating a large amount of network traffic cause short-term disruption as networks are taken offline. But they are less likely to cause lasting damage than data damaging worms, and recovery from them seems likely to be quicker and easier. Moreover, data damaging worms can *both* damage data and create a large amount of network traffic.
- **Botnets**: Worms creating botnets can cause targeted damage against specific victims. Targeted attacks could potentially cause large economic damages, but that depends to a significant extent on which victim is targeted, and there seems to be greater scope to cause harm by damaging a large number of infected victims.

[16] We discuss this in more detail in section 4 and Appendix A.8.

- **Physical damage:** Causing physical damage to operational technology, such as electrical grids or nuclear centrifuges, via cyberattacks in general and worms in particular seems very challenging (see van der Merwe et al., forthcoming). For example, Stuxnet may have been the most economically damaging publicly known cyberattack that caused physical damage. However, the total direct economic damages to Iran were only around $14M (Slayton 2017, Table 1).[17] The malware alone cost millions to tens of millions of dollars to develop,[18] and the head of the CIA at the time of the attack said that the total cost of the operation including human intelligence and building, and testing on, mock centrifuges was $1B - $2B (de Volksrant 2024). As we discuss in section 2, creating data damaging worms is 100-1,000X cheaper than this.

It is possible that worms used for state or industrial espionage could cause significant harm, though as we discuss in the next subsection, the economic costs of espionage attacks are harder to quantify and so we do not focus on them here.

Overall, we prioritised data damaging worms due to their large economic risks. However, we are open to the possibility that other types of cyberattack may pose comparable or greater economic risks. We leave this to future research.

Types Of Damage: Large individual economic events

We focus only on the economic damages caused by worm attacks (lost revenue, reduced consumption etc). However, some cyberattacks may cause other types of harder-to-quantify harm. For example, although the economic damages of the Stuxnet attack were relatively limited, the geopolitical effects may have been important. Russia's NotPetya attack on Ukraine caused substantial economic damages, but was also part of a longstanding campaign of cyber and military attacks with potentially important geopolitical consequences, which we exclude from our analysis. One of the most geopolitically costly cyberattacks was China's 2015 attack on the US Office of Personnel Management in which China stole data on everyone who had been through US security clearance up to that point. This, along with other Chinese espionage attacks, posed serious challenges to US espionage in China, Russia and Iran (Foreign Policy (2020a); Foreign Policy (2020b); Yahoo (2019)).

We focus only on easier-to-quantify economic damages because they are more tractable to analyse and rely less on access to classified information. Future work could explore other types of social damages, but they are out of scope for this report.

[17] Note that The Stuxnet worm unintentionally infected 100K systems (Falliere et al 2011, p. 5). The Stuxnet payload only activated in the Iranian nuclear facility, so the damage was likely limited, but victims would still have to pay to clean systems. I have been unable to find estimates of the collateral damage of Stuxnet.

[18] Various estimates of the cost to create Stuxnet are collated in Stuxnet development costs.

Many actors would be willing to launch worms designed to cause a large amount of economic harm, but most currently lack the ability to do so

A further reason we prioritised the worm threat model is that there is strong evidence that many actors would be willing to launch worms that could cause a large amount of economic harm if they were able to do so. Figure 1.1 below shows significant worm attacks by threat actor class, using the Johansmeyer dataset. Note that the threat actor categorisation refers to the overall capabilities of the threat actor who carried out the attack, not to the operational capacity required to execute that specific attack. For example, the attacks in 2017 were not OC4 or OC5 *operations*, but they were carried out by TA4 and TA5 *threat actors*.[19]

Figure 1.1. *Major worm attacks by threat actor (1998-2023) [Adapted from Johansmeyer, 2024]*

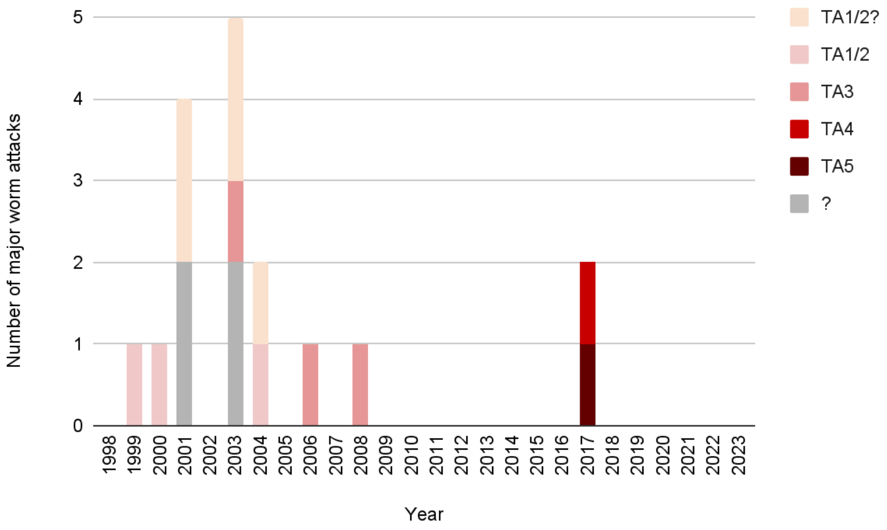


Figure 1.1 shows that:

1. The frequency of significant worm attacks has declined: Between 1998 and 2004 there were 13 worm attacks (~2/yr); between 2005-2008 there were 2 (~0.5/yr); and from 2009-2023 there were only 2 (~0.1/yr).
2. In recent years, more sophistication has been required: Prior to 2005, major worm attacks were mostly perpetrated by TA1/2 (individual hackers); from 2005-2008 there were several TA3 attacks (a group of ~10 experienced hackers); and since then the only attacks were by TA4 and TA5 actors (nation-states)

These trends were likely driven by improvements in cybersecurity:

[19] As noted above, we are sceptical of the damage estimates for many of the attacks in the Johansmeyer dataset, as they rely on damages reported in popular media sources and corporate blogs, usually without any accompanying calculations or methodology. Nonetheless, if there were highly damaging worm attacks, we would still expect them to be reported in these sorts of sources, so we think the decline in worm attacks in recent years reflects a real trend.

- Email worms became harder due to email filtering: In 2003, Microsoft introduced SmartScreen email and spam filtering technology (Microsoft 2016), which filtered >90% of unwanted spam as of 2004 (Microsoft 2004). Google (2022) now claims its filters catch >99.9% of spam.
- Improved patching: Microsoft introduced regular patching in 2003, explicitly in response to major worm attacks (DarkReading 2023). In 2004, Microsoft made "opt-in" patching the default for Windows XP, increasing the share of up-to-date users from 5% to 90% (Jenkins et al 2020, p. 4; The Register, 2005).

Thus, we can conclude from this that: (1) some actors would be willing to launch worm attacks that cause mass economic harm; but (2) lower skilled actors now lack the ability to do so, due to improved cybersecurity.

AI elite exploit development capabilities would significantly reduce barriers to data damaging worms

We argue that **if** AI gained strong elite exploit development capabilities (including both finding vulnerabilities and writing exploits of them), this would overcome the capability barriers many threat actors face to developing data damaging worms. We discuss the arguments for this in depth in section 2. Since many threat actors would be willing to release such worms, this would significantly increase the risk of worm attacks.

By contrast, other AI cyber threat models, such as espionage and critical infrastructure attacks, do not have single narrow bottlenecks, but instead require much broader AI capability improvements. These sorts of attacks involve significant planning, a large number of steps and distinct tasks, and a high degree of autonomous action and situational decision-making in novel or undocumented environments (International AI Safety Report, 2025, p.75). Thus, uplift to these sorts of attacks sets a much higher capability bar. If AI were to develop these capabilities, they may be covered by autonomy evaluations, and seem concerning primarily for reasons independent of cyber risk.

Summing up

In this section, we have clarified and justified our focus on the data damaging worm threat model. Among cyber threat models, we think it is especially concerning, though we are open to the possibility that other threat models may be more concerning and may be worth further research.

1.4. Should the risk of data damaging worm attacks be mitigated prior to deployment via AI Safety Frameworks?

As noted in section 1.1, AI companies have defined various capability thresholds in their Safety Frameworks, which determine which threats should be mitigated prior to model deployment.

Meta (2025, p. 12) and OpenAI (2025 p. 4) have argued that in order for threats to be included in their Safety Frameworks, they should meet most or all of the criteria outlined in Table 1.5.[20]

***Table 1.5**. Criteria for including risks in AI Safety Frameworks*

Criterion	Description	Explanation
Plausible	It must be possible to identify a causal pathway for a severe harm in the capability area, enabled by frontier AI.	This ensures an evidence-led approach to risk management.
Measurable	We can construct or adopt capability evaluations that measure capabilities that closely track the potential for severe harm.	This ensures that companies can reliably track the relevant capabilities, and in turn know when risk mitigation is warranted.
Severe	The outcome would have large-scale severe effects.	Focusing on severe harm is justified because some risk mitigations are very costly to companies and to society. For example, securing model weights may impose substantial productivity costs on AI companies, which would in turn reduce the economic benefits of AI models. The costs of these risk mitigations ought to be commensurate with the benefits in terms of reduced social risk: costly pre-deployment mitigations make less sense for smaller-scale risks.
Net new	The outcome cannot currently be realized as described (e.g. at that scale, by that threat actor, or for that cost) with existing tools and resources but without access to frontier AI (e.g. available as of 2021).	It is not sufficient to show that AI models could be used for damaging malicious use. One also has to show that AI would make a counterfactual difference, or would be "net new" (Kapoor et al 2024). For example, if a malicious actor could carry out the same attack for a comparable cost by using existing commonly used cyberweapons that are openly available on the internet, then AI would not increase the risk compared to the counterfactual.
Instantaneous or irremediable	The outcome is such that once realized, its severe harms are immediately felt, or are inevitable due to a lack of feasible measures to remediate.	Managing risks after model deployment better enables society to adapt and learn about model capabilities and impacts over time. For example, if AI models contribute to bias or discrimination, society can learn about such effects over time and mitigate the effects e.g. via existing anti-discrimination law. This is less viable for risks that cause instantaneous or irremediable impacts. The case for learning as we go with such risks is weaker.

We think there is a *prima facie* case that the data damaging worm threat model meets many of these criteria.

[20] Meta does not include the ‘measurability’ criterion.

1. The risk is **plausible**, as there is historical precedent for increased access to elite exploits leading to severe data damaging worm attacks (see section 2). There are also threat actors who seem willing to carry out such attacks (see section 3)
2. The relevant capabilities are **measurable**, as vulnerability discovery and exploit development are already a major focus for AI companies (see section 1.5)
3. The risk would be **net new** because many threat actors are currently unable to develop the elite exploits that could be used in data damaging worms (see section 2).
4. The economic costs could be near-**instantaneous**: a data damaging worm could cause billions to tens of billions of dollars in damage in less than a day (see section 4).
5. It is less clear whether data damaging worms meet the '**severe or catastrophic harm**' criterion; that depends on how 'severe' is defined. OpenAI (2025, fn. 1) defines 'severe' as the "death or grave injury of thousands of people or hundreds of billions of dollars of economic damage". As we discuss in Sections 4 and 6, in our view, it is unlikely that data damaging worms could meet this bar, at least in terms of expected harm.

In summary, the data damaging worm threat model meets many of these criteria, but may not meet the 'severe harm' criterion, depending on how it is defined.

1.5. Current and future model capabilities

Our argument in this report is that *if* AI capabilities uplift threat actors to find elite exploits (including both finding vulnerabilities and writing the code to exploit them), then the social costs could be large. We take no stance on if and when AI models might reach this capability level.

Vulnerability discovery and exploit development are already a major focus for AI cyber model evaluations (International AI Safety Report, 2025, pp. 74-75; Rohlf 2024; Bhatt et al 2024; Chauvin 2024). Model evaluations suggest that frontier models currently fall well-short of being able to find the elite exploits that are the focus of this report.[21] However, model evaluations sometimes under-elicit models' cyber capabilities,[22] models are improving rapidly, and future

[21] As we discuss in section 2, developing elite exploits (including both finding vulnerabilities and writing exploits of them) seems to take months of work for top tier cyber researchers, but current frontier models are unable to reliably complete cybersecurity tasks that take humans more than a few hours.

- Across different cybersecurity evaluations, Claude 3.7 Sonnet succeeded in 30% of tasks that would take humans 3-5 hours (Anthropic, 2025, p. 34-35).
- In their evaluations, PatternLabs conclude that OpenAI's o4 model "would provide only limited assistance to a moderately skilled cyberoffensive operator" (OpenAI, 2025, p. 11). o3-mini can complete tasks unguided on Cybench (a cyber model evaluation framework) that would take humans 42 minutes to solve (Cybench Leaderboard, nd). The longest task on Cybench takes humans baseliners ~25 hours to solve (Zhang et al 2025).
- Gemini 2.5 Pro Preview is able to do around 33% of tasks (Google (2025), p. 13) that would take humans 20 steps to solve (Phuong et al (2024), p. 11). Rodriguez et al (2025, p. 9, Fig. 12) state that Gemini 2.0 Flash experimental provided weak uplift for attacks using zero-day vulnerabilities and exploits, though it is unclear how Gemini 2.5 Pro performs. It is also unclear how powerful the zero-days in this model evaluation are.

[22] Project Zero (2024a) reported that improved prompting, reasoning time and scaffolding could significantly improve performance on common cybersecurity benchmarks.

capabilities are hard to predict (International AI Safety Report, 2025, pp. 74-75).[23] It is possible that models will gain the ability to find elite exploits in the near future, though we emphasise that this sets a high capability bar, one that is much higher than tested for in current model evaluations.

1.6. Methodology and report outline

In this report, our aim is to estimate the *marginal* or "net new" risk posed by future hypothetical AI capabilities, compared to existing technologies (Kapoor et al., 2024; NIST, 2025; Meta 2025, p. 12). If some threat actors can already launch a worm attack, then AI would have limited counterfactual effect on risk.

Thus, we first estimate *baseline* damages from worm attacks by different threat actors, assuming no further progress in AI. Then we estimate *marginal* damages, assuming that AI models enable different threat actors to find elite exploits. To structure our analysis, we use a simple model that breaks down the expected damage estimate into three parameters:

- The ***capabilities*** of different threat actors: For each threat actor class, the probability over the next year that a *randomly selected* threat actor in that class can launch data damaging worms if they wanted to, assuming no further improvements in AI (see Section 2). Note that we assume for ease of analysis that all threat actors in the same class have the same capability level.
- How ***willing*** threat actors are to launch data damaging worm attacks: the probability over the next year that *at least one* threat actor in each class would be willing to launch a data damaging worm attack if they were able (see Section 3).
- The ***damages*** from such attacks: how much economic harm such attacks would do (see Section 4).

We focus only on expected damages from data damaging worms that could, on their own, cause >$1B in damage. So, we do not try to estimate the cumulative expected damages from all data damaging worms, including smaller attacks. Firstly, this simplifies the analysis. Secondly, because worms spread exponentially, we think most of the expected damages are from these tail events.

We can bring estimates of the three parameters together to calculate baseline and marginal expected damages from data damaging worm attacks launched by different threat actors. (see Section 5). The formula for baseline expected damages is shown below:

Baseline **expected damage from threat actor**$_i$ = Capability$_i$ * Willingness$_i$ * Damages

[23] In 2024, Google's Project Zero (2024b) claimed that an LLM agent had found what they believe "is the first public example of an AI agent finding a previously unknown exploitable memory-safety issue in widely used real-world software". However, they note that "it's likely that a target-specific fuzzer [a commonly used tool for finding vulnerabilities] would be at least as effective (at finding vulnerabilities)." It is also unclear how exploitable this vulnerability is, and it seems to fall well-short of being an elite exploit.

The formula for marginal expected damages from AI uplift is:

> ***Marginal* expected damage assuming maximal AI elite exploit uplift to threat actor$_i$**
> = ((Capability$_i$ | AI uplifts threat actor$_i$ to find elite exploits) * Willingness$_i$ * Damages * Adjustment for the effect of AI on defence) - Baseline damages$_i$

Three things should be noted about this formula. First, the effect of AI on attacker capability is captured by conditioning the probability that attackers can launch data damaging worms on AI uplifting threat actors to find elite exploits. Second, this formula includes an adjustment for how AI benefits defenders. The AI capabilities we investigate here - vulnerability discovery and exploit development - are *dual use* in that they are useful to both attackers and defenders. It is important to consider how benefits to attack and defence net out. Third, to capture the marginal impact of AI, we subtract baseline damages.

Our ultimate aim in this report is to generate an estimate of baseline and marginal damages that is well-supported by evidence. To that end, our approach to estimate expected costs involved:

1. Empirical research: We reviewed case studies of past worm attacks, and gathered information relevant to the parameters in our model in order to produce our own initial estimates.
2. Expert feedback: We sought expert feedback from cyber and national security experts who provided comments on earlier drafts.
3. Expert survey: We surveyed 8 subject-matter experts and 13 credentialed ‘superforecasters’ on the parameter inputs into the model, and other questions relevant to the report. Given the small sample, the results of the survey should be interpreted with caution, but it provides initial information on a wider range of perspectives on the risk. Future work could expand the survey to produce more robust consensus estimates. We will discuss the survey methodology and results in depth in follow-up work.

We believe our approach of decomposing the final expected damage estimates into distinct subquestions is likely to produce more accurate answers than directly estimating the final estimate (Tetlock, *Superforecasting* (2015), Ch. 5).

In Section 6, we briefly discuss one approach to mitigating the risks of elite exploit discovery and data damaging worms: giving defenders early access to models. This approach to risk mitigation may be worth further research.

2. How *capable* are threat actors of releasing data damaging worms at present?

In this section, we first review the evidence that if threat actors had access to elite exploits, this would make it significantly easier to create data damaging worms. We argue that elite exploits seem especially useful for data damaging worms, and explore case studies of the WannaCry and NotPetya, each of which seemed to be enabled by the public leak of elite exploits.

We then discuss how capable different threat actors are of developing elite exploits and releasing data damaging worms. We discuss evidence around how hard it is to create elite exploits, and which actors are capable of doing so. We then outline expert estimates of the probability that different threat actors could develop elite exploits and release data damaging worms over the next year, assuming no further improvements in AI.

2.1. Elite exploit development as key capability

2.1.1. Hypothesis

The hypothesis we defend in this section is as follows:

> **If** AI enables different threat actors to develop elite exploits (including both finding vulnerabilities and writing the exploits of them), then this would make it substantially easier for many threat actors to create data damaging worms that could cause severe economic harm (>$1B in damage). Creating elite exploits is substantially harder than all of the other tasks involved in developing data damaging worms combined. Thus, AI uplifting threat actors to find elite exploits would substantially lower barriers to the creation of data damaging worms for many threat actors.

Since, as we argue in section 3, many threat actors appear willing to release worms that cause severe economic harm, AI elite exploit development capabilities would therefore substantially increase the risk of data damaging worms actually being released.

However, it is less clear whether elite exploits are strictly *necessary* for data damaging worms; it is less clear whether they are true ‘bottleneck’ in this sense. It might be that data damaging worm attacks that do *not* use elite exploits could also comparably cause severe harm. Expert reviewers have mentioned several possibilities:

1. The combination of numerous weaker exploits of less widely used and less well-defended software in one worm. The ease of finding weaker exploits would be offset by the extra costs involved in finding a larger number of exploits, and combining them in a single worm.
2. Worms that require some user interaction to spread but are still able to damage data on a large number of systems. As discussed in section 1, many non-zero-click worms

released prior to 2008 nevertheless spread very widely. Although email worm attacks now seem less viable, other non-zero-click worms may still be able to spread widely. For example, 'watering hole' attacks that infect systems when users visit popular websites, require users to take specific actions, but may nevertheless spread widely.
3. Agentic 'polymorphic' worms which can, unlike the worms we consider here, autonomously change their code after release in order to circumvent defence.

These threat models may justify different cyber capability thresholds than the one we discuss here, and these other capability thresholds may be triggered earlier than elite exploit development. These threat models may warrant further research, but are out of scope for this report.

This being said, we still think it is plausible that it is much harder to create data damaging worms that cause severe economic harm without access to elite exploits. Indeed, it is notable that perhaps the most damaging worms ever - WannaCry and NotPetya - each used elite exploits.

Whether or not elite exploits are necessary for data damaging worms, it is not relevant to the main hypothesis we defend in this section: that AI-enabled elite exploit development would substantially lower the capability barriers to the creation of data damaging worms.

In the next two sections, we present two arguments for this hypothesis.

1. The features of elite exploits seem especially well-suited to data damaging worms that can cause severe economic harm.
2. Case studies of the WannaCry and NotPetya worm attacks.

2.1.2. Features of elite exploits seem especially well-suited to data damaging worms

To recap, elite exploits are defined as follows:

Table 2.1. *Defining features of elite exploits*

Exploit feature	**Definition**
Zero-click	Infection requires no user interaction, such as opening emails, clicking links or visiting a webpage.[24]
Remote code execution	Allow attackers to execute arbitrary code on a system without the user's knowledge, and without attackers requiring physical access to the system

[24] Note that on some definitions of 'zero-click', not all zero-click exploits require absolutely no user interaction to spread. For example, some watering hole attacks can infect a system if a user visits a website but does not click on any links on the website. Some definitions class this as a zero-click exploit (e.g. SmarterMSP 2024). On our definition, these would not be elite exploits because they require the user to visit a compromised website, and therefore involve user interaction.

High privileges	Privileges are the permissions granted to users, programs, or processes to perform specific actions on a system or access particular resources. Privileges levels range from (low to high): sandboxed application, user, administrator, to system level. Elite exploits have administrator privileges or higher.
Targets widely used software	Effective against >10M systems

Exploits with these features are especially useful for data damaging worms that can cause severe economic harm.

No user interaction

Exploits which require no user interaction enable autonomous spread. If a user needs to e.g. click a link, open an email, or visit a specific website in order to be infected, the worm is likely to spread more slowly and less widely because it would rely on users taking certain specific actions that a substantial fraction of users are unlikely to take. Moreover, defenders can respond reactively to warn users about which actions to avoid after the worm has been released.

Remote code execution

Remote code execution is important because it allows attackers to deploy implants and payloads without already having local access to the network, which is necessary for widespread propagation, and for the deployment of data damaging payloads.

High privileges

High privileges (i.e. admin or higher privileges) are important for (1) causing damage to infected systems and (2) for propagation.

Damage

In general, the higher the privileges malware has, the more damage it can do to infected systems (Devicie nd). If malware has privileges below the user level, e.g. corrupts a sandboxed application, there is much less scope to cause significant damage by encrypting important files. In order to cause significant damage, it seems plausible that malware must at least have user-level privileges, as this allows access to user files that can then be encrypted (Engage Employee nd). However, malware with only user-level privileges might not be able to encrypt local or cloud backups. Typically, administrator or domain admin privileges are required to do this, and attackers often try to escalate privileges to administrator or higher (Red Canary 2019; BleepingComputer 2019).

For example, because NotPetya had system-level privileges on infected systems, it was able to encrypt and corrupt not only the data files but also the Master Boot Record and Master File Table, which made it very difficult or impossible to restore the affected systems to a usable state (Forbes 2017; Malwarebytes 2017).

Propagation

In general, the higher level of privileges a worm has, the easier propagation will be within *local* networks. In order to spread within local networks, attackers often seek to gain administrator privileges or higher (Microsoft 2022). Indeed, NotPetya spread in part by gaining admin privileges on local area networks (Altaro 2021). In targeted ransomware incidents, attackers often try to gain Domain Admin privileges in order to push ransomware to all endpoints or devices in a network (Microsoft 2022; SentinelOne 2022). Gaining high privileges is often useful for spreading between networks or between cloud environments, depending on the nature of the malware and the systems it is targeting (Cado Security nd; Aquasec 2023; BeyondTrust 2017).

Widely used software

We stipulate that elite exploits are effective against software with more than 10M users.
The basic reason for this is that worms with large reach can do more damage. As we discuss in section 4, WannaCry and NotPetya caused ~$1B and ~$10B of damage after spreading to hundreds of thousands of systems. This might be thought to suggest that focusing on worms with much more limited reach could be justified, as these worms could still cause >$1B in damage.

However, capability thresholds should be determined by the *expected* damages of risks. Consequently, the actual damages that a worm attack would cause, if it were to occur, should be discounted by:

- The probability that the threat actor can complete the other steps involved in making the worm.
- The probability that threat actors are willing to release the worm.
- The probability that the worm is stopped before it infects the total pool of vulnerable hosts. For instance, were it not stopped after 7 hours, WannaCry could potentially have infected tens of millions of systems, but in fact only infected hundreds of thousands.[25]

These factors would make the expected damages from data damaging worms much lower than the actual damages if they were to occur. So, we focus on worms with much greater reach.

2.1.3. Case studies: elite exploits enabled the WannaCry and NotPetya attacks

Since 2009, the only significant attacks were WannaCry and NotPetya in 2017. These attacks were the product of an unusual natural experiment in which elite exploits, including the EternalBlue exploit, initially developed by the NSA were leaked to the public in April 2017. North Korea and Russia then used these elite exploits in the WannaCry and NotPetya worms 1-2 months later.[26]

[25] See section 4.

[26] For more detail on how WannaCry and NotPetya worked, see Appendix A.3.

The EternalBlue exploit

EternalBlue directly exploits vulnerabilities in a network-facing protocol - the Server Message Block v1 (SMBv1) Protocol, a legacy file sharing protocol used in Windows systems up to 2017. The SMBv1 protocol exploited by EternalBlue was first developed in 1983 (Avast 2020), and is now widely recognised as insecure (Microsoft 2025). On many systems vulnerable to EternalBlue, port 445 (over which SMBv1 communicates) was exposed to the open internet, so a network scanning feature could propagate the exploit to other systems (Upguard 2025).

EternalBlue was unusually powerful tool for a worm attack (Microsoft 2023; SentinelOne 2019):

- Because it exploited a network-facing vulnerability, it could spread easily across exposed internal and external networks.
- It exploits these vulnerabilities without requiring prior authentication or privileges. EternalBlue was therefore also a *zero-click* exploit in that it could infect systems without user interaction.
- It enabled *remote code execution* on the infected system, i.e. attackers could execute arbitrary code on the infected system, allowing them to deploy further implants and payloads to enable further spread and damage on the infected system.
- It gained *system level* privileges (the highest level of privileges on Windows systems), which meant that it had access to system-level files, including backups. This increased the potential damage a worm using EternalBlue could do.
- At the time, SMBv1 was a ubiquitous protocol on Windows systems. Prior to the release of the patch for the vulnerabilities exploited by EternalBlue in March 2017, ~400 million systems were vulnerable to it (Coburn et al (2019), p. 44).

The leak of EternalBlue and other elite exploits quickly led to the WannaCry and NotPetya attacks

The NSA developed EternalBlue in 2012 at the latest, and used it in surveillance and counterterrorism operations for at least five years (NYT 2019). EternalBlue and a trove of other NSA hacking tools were stolen by a hacker group known as the Shadow Brokers, and sold or leaked publicly from summer 2016 to spring 2017, which in turn led to the WannaCry and NotPetya worm attacks. Figure 2.2 outlines the timeline leading up to WannaCry and NotPetya:

Figure 2.2. Wannacry and NotPetya timeline (based on public reporting)

- **2012**: EternalBlue was first developed by the NSA by 2012 at the latest, and was used for targeted espionage for years (NYT 2019).
- **Pre-2017:** At some point prior to 2017, the Shadow Brokers stole various NSA hacking tools, including EternalBlue (EFF 2016; NPR 2017).
- **February 2017:** Russian hackers responsible for the NotPetya attack may have had access to the Shadow Brokers exploits (Ars Technica 2017).
- **Early 2017**: The NSA informs Microsoft of the vulnerabilities possessed by the Shadow Brokers (Ars Technica 2017)

- **14th March 2017**: Microsoft released a patch for the vulnerabilities (Microsoft 2017), though many users delayed deploying or installing the patch.
- **14th April 2017**: The Shadow Brokers leaked EternalBlue, along with other stolen NSA hacking tools (Ars Technica 2017).
- **12th May 2017**: WannaCry attack, using EternalBlue, launched (Wired 2017). A cybersecurity researcher discovered a kill switch for the worm so that after 7 hours, no new systems were encrypted and spread was slowed significantly (Kryptos Logic (2017); Malwaretech 2017; Cisco 2017).
- **27th June 2017**: The NotPetya attack, using EternalBlue and EternalRomance (another NSA elite exploit), launched (ESET 2017).

How elite exploits were used in the WannaCry and NotPetya attacks

WannaCry and NotPetya used EternalBlue in different ways.

How WannaCry worked

1. ***Initial infection:*** Used EternalBlue to infect vulnerable systems.
2. ***Propagation to other systems:*** WannaCry contained a network scanner which searched for other vulnerable machines on the local network and at random IP addresses on the internet (Nguyen et al 2024; Microsoft 2017). Once a vulnerable system was found, WannaCry used EternalBlue to exploit it in the same way it infected the first machine (TrendMicro 2017). The malware would deliver its payload to the newly compromised machine, effectively turning it into another worm node. This self-replication process continued indefinitely, rapidly spreading the infection across systems.
3. ***Installation of backdoor:*** EternalBlue installs a backdoor called DoublePulsar (another stolen NSA hacking tool) on the infected system. This allows persistent access and remote code execution and system level privileges, which in turn allows the installation of the payload (Microsoft 2017; Threat Post 2017).
4. ***Installation of ransomware:*** Wannacrypt ransomware is installed by DoublePulsar (Kaspersky 2022). It encrypts files on the infected system and displays a message asking the victim to pay ransom in order to decrypt their data (Microsoft 2017).

NotPetya was designed to limit damage to Ukraine, and so was designed differently to WannaCry.

How NotPetya worked

1. ***Initial infection:*** The Russian attackers compromised M.E.Doc accounting software that was used by 80% of domestic firms in Ukraine in a ‘supply chain attack’ (Reuters 2017; Ars Technica 2017; Cisco 2017).
2. ***Propagation to other systems:*** Unlike WannaCry, NotPetya did not spread *between* networks, but only *within* them (Cybereason nd). Consequently, spread would be limited to organisations that used the Ukrainian M.E.Doc accounting

software.[27] NotPetya used EternalBlue, and another NSA elite exploit called EternalRomance,[28] to target systems that had not installed the patch released by Microsoft months earlier. NotPetya could also infect *patched* systems because it used Mimikatz, an openly available tool that could pull passwords out of memory and use them to hack into other machines. With those credentials, NotPetya could use legitimate system tools to move laterally within networks (Microsoft 2017; Wired 2017; Wired 2018; Dark Reading 2020).

3. ***Installation of wiper:*** NotPetya installed a wiper payload encrypted files on infected systems. Although it displays a ransomware message, this was likely a false flag to make the attack look like cybercrime. It was not possible to decrypt the data (Securelist 2017; Virsec 2017).

We think that are two potential interpretations of how elite exploits enabled the WannaCry and NotPetya attacks:

1. Russia and/or North Korea wanted to carry out a worm attack but did not have elite exploits, so the Shadow Brokers leak overcame the main technical bottleneck for them.
2. Russia and/or North Korea was able to develop elite exploits in-house, but these were useful for other purposes (e.g. espionage), so the opportunity cost of burning them in worm attacks was high. But the opportunity cost of using the Shadow Brokers exploits was lower because they were leaked publicly and so would soon be patched anyway.

We are unsure which interpretation is more plausible. Our analysis suggests that Russia was probably able to develop elite exploits, but this seems less likely for North Korea. In any case, both interpretations suggest that elite exploits are in short supply even for OC4 and OC5 state actors.

Developing EternalBlue required ~10X as much skilled staff time as the other steps involved in the attacks

We think it is plausible that developing EternalBlue was much harder than the other steps involved in creating WannaCry and NotPetya. The limited public evidence suggests that creating EternalBlue took ~10X as much skilled hacker time as the other steps involved in making the worm.

Wiper or ransomware payloads comparable to those used in WannaCry and NotPetya typically sell for ~$1K (Venafi 2022). All the other software used in the attacks was free. Thus, the vast majority of the cost of developing the worms was from skilled labour.

[27] The worm was apparently not intended to spread beyond Ukraine, but it nevertheless did, as some companies had Ukrainian subsidiaries. Indeed, the worm unintentionally hit many Russian companies (Greenberg, *Sandworm*, p.198).

[28] EternalRomance is another exploit of Windows SMBv1 vulnerabilities, but targets older versions of Windows (Crowdstrike 2017).

According to public reporting, “[NSA] analysts spent almost a year finding a flaw in Microsoft’s software and writing the code to target it” (NYT, 2019), whereas WannaCry was released one month after EternalBlue was leaked.

However, it is unclear how many people were involved in either effort. The US has indicted three North Koreans for making WannaCry (DoJ 2021), though more may have been involved in development.[29] Assuming the same number of cybersecurity analysts were involved in each of (1) developing EternalBlue, and (2) writing the WannaCry worm using it, this suggests that (1) took ~12X as much staff-time as (2) (12 months vs 1 month).[30] Assuming the NSA and North Korean analysts were similarly skilled, this suggests that creating an elite exploit comparable to EternalBlue took ~10X as much skilled labour as writing a data damaging worm using it.[31]

Our own rough estimates suggest that WannaCry and NotPetya cost on the order of $100K to develop in US-equivalent hacker costs, once the actors had access to the NSA tools.[32]

[29] One expert told us that making the worm from the exploits would be within the reach of 2-3 not particularly experienced hackers.

[30] The NotPetya case is less comparable here because of the different aims and design of the malware. The Russian attackers may have had access to the NSA tools four months prior to, though part of this time may have been spent carrying out the supply chain attack on M.E.Doc, the Ukrainian accounting software.

[31] It seems likely that the NSA analysts were more skilled than the North Korean hackers, so the skill-adjusted time for EternalBlue was likely higher than this.

[32] See Development costs WannaCry and NotPetya .

The other steps involved in making the worms still seem non-trivial

Although developing WannaCry and NotPetya was much easier than developing elite exploits, there is some evidence that developing them still required significant skill.

- **Both attacks were carried out by TA4 and TA5 state actors** rather than TA1/2 lone wolf hackers, even though there are far more TA1/2 actors than TA4/5 actors, and some TA1/2 actors would likely be willing to launch such an attack. This suggests that most or all non-state actors who would have released a worm as damaging as WannaCry (if they could) were unable to do so in the 1-2 month period after the Shadow Brokers exploits became available.
- **The WannaCry code contained numerous errors** (Ars Technica 2017; WIRED, 2017a; WIRED 2017b), even though it was developed by state-level hackers. (By contrast, experts suggested that NotPetya was sophisticated and well-tested (Ars Technica 2017; Lawfare 2017).)

This suggests that even given access to an elite exploit like EternalBlue, it was difficult for non-state actors to develop a worm using it before most systems were patched. Thus, if AI only has strong elite exploit capabilities, but doesn't help actors write worms, the latter step could still be a meaningful bottleneck for lower skilled actors.

It is difficult to estimate how easy it would be for non-state hackers (TA1-TA3) to make a worm as damaging as WannaCry, once they had access to the elite exploits. Table 2.3 discusses case studies of the time taken to develop different worms, which helps to shed light on this question.

Table 2.2. *Case studies of development times for different worms*

Worm	Year	Threat actor	Description	Implied non-exploit development time
Conficker	2008	TA3	Five hackers capable of OC3 operations developed five major variants over 6 months (ACM 2009; DailyKos 2009). One variant modified 85% of the original code (SRI 2009). However, Conficker used elite exploits, so it is not clear how comparable this is to the time taken to develop worms conditional on access to elite exploits.	1 month of OC3-level hacker time per variant.
Mirai	2016	TA2?	Three hackers developed over ~3 months (Wired 2023). Created a large botnet of poorly defended internet of things devices. The skill required may be comparable to the skill required to develop a worm conditional on access to elite exploits.	9 months of OC2-level hacker time.
WannaCry	2017	TA4	North Korea developed WannaCry one month after the release of EternalBlue. At least 3 hackers involved in development (DoJ 2021).	~3 months of actors capable of OC4 operations
NotPetya	2017	TA5	Developed in 2-4 months by at least 4 hackers (DoJ 2020) by Russia, which is capable of OC5 operations.	8-16 months of actors capable of OC5 operations
EternalRocks	2017	TA2?	EternalRocks used seven of the leaked NSA hacking tools, compared to two used by WannaCry, and some argued that it was more complex than WannaCry, though less dangerous because it did not have a destructive payload (BleepingComputer 2017). It was apparently developed by one hacker 1 month after the Shadow Brokers leak. It is unclear what skill level the hacker had. The author shut it down within a few weeks of release (BleepingComputer 2017).	1 month of hacker time. Skill level unknown.

We discuss how these considerations influence our working assumptions about actor capabilities in Section 2.2.2.

It is worth noting that it might be the case that an AI that is good at developing elite exploits would *also* be good at writing data damaging worms because both writing exploits and writing worms may be downstream of the general coding abilities of AI models. Thus, if AI does uplift TA1-3 actors to find elite exploits, it might also uplift them to write data damaging worms. Whether or not this is true has important implications for the expected damages implied by this threat model.

2.2. Assessing how capable different threat actors are of finding elite exploits

In the previous section, we argued that if actors had access to elite exploits, this would substantially lower the capability barrier to the creation of data damaging worms that could cause severe economic harm. In this section, we present evidence on the probability that threat actors can already develop elite exploits (including finding vulnerabilities and writing the code to exploit them). We first present a range of evidence suggesting that it is very hard to develop elite exploits (section 2.2.1), and then discuss the current capabilities of different groups of threat actors.

2.2.1. It is very hard to develop elite exploits

In addition to the case studies discussed above, here we discuss other evidence on how hard elite exploits are to develop. We conclude that this capability is beyond the reach of TA1/2 actors, but TA5 actors are clearly capable. The capabilities of TA3 and TA4 actors are less clear.

Elite exploits are expensive, indicating they are hard to develop

There are various different markets for exploits, including:

- **Bug bounties** offered by software developers for white hat hackers to find vulnerabilities in their systems.
- **Grey market brokers** like Zerodium and Crowdfense buy the code for exploits from individuals, groups or companies and then sell them to governments.
- **Commercial surveillance vendors** like NSO Group and Intellexa sell spyware using elite exploits to state intelligence agencies, who then use them for targeted espionage.

Table 2.4 summarises the prices in these different markets.

Table 2.3. *Elite exploit prices in three different markets*

Market	**Est. Price ($M)**
Bug bounties	$100K-$1M per zero-click RCE exploit w/ kernel privileges for individual device or system.[33] (The price for an exploit effective against numerous different systems would be higher)
Grey market brokers	$1M-$9M per elite exploit (These prices are likely below true buyer willingness to pay due to adverse selection).[34]
Commercial surveillance vendors	$100K-$800K per *device* infected by a tool including zero or one-click exploit + spyware.[35] (The same tool would be sold to numerous customers, and would be used by each customer to target numerous (10 or more) devices. This suggests that the market value of the elite exploits + spyware would be at least millions of dollars, though it is hard to know how much of the value derives from the elite exploits vs the spyware.)

Overall, we think the prices for exploits in these different markets suggest that *elite* exploits, as defined here, would be worth on the order of $10M. Moreover, prices in these markets have been rising above inflation over time, which suggests that it is becoming harder to find elite exploits due to improved cybersecurity (Tech Crunch 2024).[36] The fact that many organisations sell elite exploits, or malware tools using them, to state intelligence agencies also provides further evidence that elite exploits are difficult to develop, even for states.

Since 2020, known elite exploits were developed by TA4 and TA5 threat actors or specialized groups

There are no public comprehensive data on the number of elite exploits developed since 2017 and who developed them. This is in large part because many elite exploits may be used in secret for years before ever becoming publicly known (e.g. the NSA used EternalBlue secretly for at least five years).

[33] Apple offers $100k to $1M for zero-click remote access exploit chain with full kernel execution and persistence (i.e. the exploit continues to work after the device has been rebooted) on a single recently released device. Google offers $1m for a zero-click RCE with persistence exploit that is effective against all vulnerable builds and models of Pixel Titan M. The price for vulnerabilities affecting a large number of distinct systems would be much higher. WannaCry was effective against ~400M systems at the time of the attack. This suggests that in today's prices, EternalBlue would be worth millions of dollars, and plausibly on the order of $10M.

[34] Smeets (2022) argues that "The zero-day exploit market is a market with extreme information asymmetries. The seller has much more information about whether the exploit is actually working. The market is also flooded with lemons. Many of the exploits offered are a lot less reliable than sellers initially report. Also, the buyer of an exploit is not always able to test the exploit before purchasing it, as the economic value would be lost once given to the buyer for "testing." This structural setup makes even beneficial zero-day transactions difficult." If buyers are unable to tell the difference between strong and weak exploits, they would be unwilling to pay high prices. Consequently, the price will be lower than what sellers of high-quality exploits would sell for, driving them out of the market.

[35] Commercial Surveillance Vendor product offerings prices are collected together in: ⊞ Exploit prices from commercial surveillance vendors

[36] Prices on exploit broker platforms have increased in recent years: in 2019, the highest bounty offered by Crowdfense was $3M (Tech Crunch 2024), whereas today the highest is $9M.

Google's Threat Analysis Group maintains a database of zero-day exploits that were publicly discovered being used in cyberattacks (e.g. for espionage or for financial crime). It is not straightforward to analyse which of these exploits are elite-level,[37] but our own shallow analysis of this dataset suggests that from 2020-24 around 4 were elite-level, or one per year (see Appendix A.6)). These were developed by the US and its allies (in the TA5 threat actor class), the commercial surveillance vendor NSO, and a white hat hacker called Orange Tsai. NSO Group is plausibly TA3 or TA4.[38] Orange Tsai has won numerous hacking competitions, and one expert told us that he is one of the best white hat hackers in the world, and he.[39]

As mentioned, this is likely an underestimate of the number of elite exploits developed each year. We may have missed some known cases and we would not be surprised if zero-day exploits *discovered* being used were only a small fraction of the total actually developed in a year. Nevertheless, the rarity of publicly discovered zero-day elite exploits is also evidence that they are out of the reach of lower skilled actors who might use them for criminal or destructive purposes, as such attacks would quickly become publicly known.

It is also notable that more recent zero-day elite exploits are more complex than older elite exploits.[40] This is further evidence that elite exploits are becoming harder to find over time due to improved cybersecurity.

Stuxnet development costs

The total costs for the whole Stuxnet operation were estimated to be \$1B-\$2B, but this includes the costs of building mock centrifuges and testing the Stuxnet malware on them, as well as extensive human intelligence (Dark Reading 2024; de Volksrant 2024). The Stuxnet worm used an elite exploit chain of multiple exploits, including numerous zero-days.[41] There are a range of

[37] Google Threat Analysis Group (2023) states that no 0-click exploits were publicly detected and disclosed being used in cyberattacks in 2022, but that some exploits for the installation of NSO's Pegasus spyware were found in 2021. (In reports for other years, it does not state how many zero-click exploits were publicly discovered being used for cyberattacks.)

[38] They have 750 staff (Fortune 2021), many of whom are drawn from Israeli intelligence (NYT 2022), and they specialise in selling spyware using powerful exploits to state intelligence agencies (Google Threat Analysis Group, *Buying Spying*, 2024)

[39] He has won awards including "Master of Pwn" at Pwn2Own 2021 and 2022. His research earned him the Pwnie Awards winner for "Best Server-Side Bug" in 2019 and 2021 and also secured 1st place in the "Top 10 Web Hacking Techniques" for 2017 and 2018 (bio).

[40] EternalBlue was a single exploit that directly enabled remote code execution with kernel privileges. Today, due to improved cybersecurity, chains of 3-4 exploits would likely be required to have the same functionality (Google, *Buying Spying*, (2024, p. 29)). Due to measures such as sandboxing and process isolation, even if attackers find a way to run their code via a malicious attachment or a compromised app, they will only have limited access and privileges. Consequently, additional exploits (enabling sandbox escape exploits and privilege escalation) would also be needed to gain the functionality of elite exploits. For an example of this type of elite exploit chain and discussion of how it could be used in a worm, see Appendix A.5.

[41] Sources differ on the number of exploits used in the Stuxnet worm. Falco (2012, Table 1) states that there were six, including 5 zero-days, while Falliere et al (2011, Table 2) states that the final version of the Stuxnet worm used 7 exploits, including 6 zero-days.

estimates for the costs to develop the Stuxnet malware and exploits, not including the other costs of the operation (Falco 2012, pp. 20-21; Slayton 2017, pp. 99-102; Symantec 2010).[42]

- Various sources suggest that the malware and exploits for the Stuxnet worm cost $3M to $20M to develop in 2005 dollars.
- Various sources suggest that developing the Stuxnet worm took months to dozens of person-years for cybersecurity professionals.

This is further evidence that developing effective worms is very costly, though the Stuxnet worm also had substantially different functionality and aims to the WannaCry and NotPetya worms, so the comparison is not straightforward.

2.2.2. Estimating actor capabilities

In light of the discussion in this section, we now develop estimates of the probability that different threat actors can: (1) develop elite exploits (including finding vulnerabilities and writing the code to exploit them); and (2) conditional on having access to elite exploits, can develop a data damaging worm using them.

Table 2.4 summarises our estimates of the probability that different groups of threat actors can already develop elite exploits, and provides supporting arguments for these estimates. To be precise, we estimate the probability that *any randomly selected* threat actor in a threat actor class is able to develop elite exploits. For ease of analysis, we assume that all threat actors in each class have the same capability level. We updated some of these estimates following the results of the expert and superforecaster survey.

Table 2.4. Rationales for our current best guess for the annual probability that any randomly selected threat actor in a given OC subset can develop elite exploits

Threat actor	Rationale	Elite exploit capability
TA1	● It plausibly costs >$1M to develop elite exploits. TA1 actors have at most a budget of $1K for a specific attack. This suggests that it is extremely unlikely that they can develop elite exploits. ● By our analysis, no elite zero-day exploits discovered being used in cyberattacks since 2020 were developed by TA1/2 actors. ● States pay millions of dollars for elite exploits which suggests that they are difficult to develop even for states	Very remote chance (~0%) *Significant uplift needed*
TA2	● It plausibly costs >$1M to develop elite exploits. TA2 actors have at most a budget of $10K for a specific attack. This suggests that it is extremely unlikely that they can develop elite exploits. ● See TA1 considerations	Very remote chance (~0%) *Significant uplift needed*
TA3	● Elite exploits cost >$1M to develop and TA3 actors have a budget of up to $1M for a specific attack, which suggests that they may have the budget required to	Very unlikely (1-10%)

[42] These estimates are collated in this sheet Stuxnet development costs .

	develop elite exploits. ● Of elite zero-day exploits discovered being used in cyberattacks since 2020, some could arguably be classed as being developed by TA3 actors (e.g. NSO Group). ● 2008 Conficker worm used sophisticated elite exploits and was built by TA3 actor, though this is likely harder now. ● States pay millions of dollars for elite exploits which suggests that they are difficult to develop even for states	*Significant uplift needed*
TA4	● TA4 actors have $10M for a specific attack and staff of 100 cybersecurity professionals. This level of resources seems sufficient to develop elite exploits. ● The WannaCry example is some evidence that North Korea could not develop elite exploits in-house. ● North Korea has been discovered using zero-day exploits in cyberattacks (Google (2024) p. 14), but from the cyber case studies we have looked at, we are not aware of any known cases in which North Korea has used *elite* exploits.	Realistic possibility (5–60%) *Some uplift needed*
TA5	● TA5 actors have 1K top tier staff and $1B for a specific attack. Seems clearly sufficient to develop elite exploits. ● Most known elite exploits have been developed by OC5 actors. US and its allies used elite exploits in EternalBlue espionage, Duqu, Flame and Stuxnet. Assuming China's cyber capabilities are comparable to the US,[43] it is very likely that China is also able to develop elite exploits.[44]	Almost certain (90-100%) *No uplift needed*

Table 2.5 below summarises our estimates of the probability that different threat actors can already develop data damaging worms that cause severe economic harm, conditional on them having access to elite exploits. It also provides supporting arguments for these estimates.

Table 2.5. Rationales for our current best guess for the annual probability that any randomly selected threat actor in a given TA class can develop data damaging worms assuming they have access to, but did not necessarily develop, elite exploits

Actor type	Rationale	Elite exploit capability
TA1	● The estimated costs to complete the non-exploit development tasks involved in WannaCry and NotPetya were ~$100K, which is 100X TA1 budget for an attack. ● Significant worm attacks using the Shadow Brokers tools were not launched by TA1 and TA2 actors, despite apparent willingness to launch such attacks. Though such attacks would only have been effective within around 3 months of the Shadow Brokers leak, as patches would have been deployed. So, it is	Extremely unlikely (0.1%-2%) *Significant uplift needed*

[43] RAND (2021) estimates that the Department of Defense has ~50,000 cyber staff (see footnote on p. 48), though many US cyber staff are outside the DoD. Christopher Wray, the former director of the FBI, told Congress in 2023 that Chinese hackers outnumber the FBI's cyber staff 50 to 1 (CNBC 2023), though note that the FBI is only one part of the US government's cyber force. Mandiant (2013) estimated that China has "130,000 personnel divided between divided between 12 bureaus (局), three research institutes, and 16 regional and functional bureaus". USCC (2022), (p438) notes "the PLA reportedly has as many as 60,000 cyber personnel that could support cyberwarfare missions" – however this is just one arm of China's cyber army.

[44] Note that "According to publicly available reports, China stood up an elite corps for cyber operations in 1997 and established a battalion-sized information warfare unit in 2000." (USCC 2022, footnote on p. 431)

	difficult to rule out TA1&2 capability over a year.	
TA2	● See TA1 considerations ● The estimated costs to complete the non-exploit development tasks involved in WannaCry and NotPetya were ~$100K, which is 10X TA2 budget for an attack. ● The Mirai (likely TA2) authors were able to make a worm leveraging exploits of weak systems within 9 person-months. This may be comparable to the resources required to develop a data damaging worm, given access to elite exploits. ● The EternalRocks example suggests that someone was able to make a worm that was in some ways more complex than WannaCry within two months. However, it is unclear what skill level the EternalRocks author had.	Unlikely (1-20%) *Significant uplift needed*
TA3	● TA3 actors were able to develop one substantially different Conficker variant per month. ● Estimated costs to develop data damaging worms, assuming access to elite exploits, are within reach of TA3 actors. ● NotPetya case suggests that creating a well-functioning data damaging worm may have taken 8-16 months of top tier hacker skill, which is plausibly within reach of TA3 actors. ● Significant worm attacks using the Shadow Brokers tools were not launched by TA3 actors, though this may reflect limited willingness to launch such attacks.	Realistic possibility (5-60%) *Some uplift needed*
TA4	● WannaCry was released one month after Shadow Brokers leak, which suggests North Korea and other TA4 clearly has the capability over a year. ● Estimated costs to develop worms, assuming access to elite exploits, are well within the reach of TA4 actors.	Almost certain (~100%) *No uplift needed*
TA5	● NotPetya was released by Russia 2-4 months after access to NSA tools, which suggests clearly within reach for TA5 actors over a year. ● TA5 actors have developed numerous effective worms for espionage and sabotage, which suggests that they clearly can also develop data damaging worms. ● Estimated costs to develop data damaging worms, assuming access to elite exploits, are well within the reach of TA5 actors.	Almost certain (~100%) *No uplift needed*

Table 2.6 below collates these estimates and computes the “end-to-end capability” of different threat actors: the probability that they can already *both* (1) create elite exploits and (2) develop data damaging worms using those elite exploits.

Table 2.6. Collated capability estimates for different threat actors

Actor type	**Exploit Capability:** Probability an actor can succeed already over a year	**Post-Exploit Capability:** Probability an actor can succeed already	**End-to-End Capability:**
TA1	Very remote chance (~0%) *Significant uplift needed*	Extremely unlikely (0.1%-2%) *Significant uplift needed*	Very remote chance (~0%) *Significant uplift needed*
TA2	Very remote chance (~0%) *Significant uplift needed*	Unlikely (1-20%) *Significant uplift needed*	Very remote chance (~0%) *Significant uplift needed*
TA3	Very unlikely (1-10%) *Significant uplift needed*	Realistic possibility (5-60%) *Some uplift needed*	Very unlikely (1-10%) *Significant uplift needed*
TA4	Realistic possibility (5–60%) *Some uplift needed*	Almost certain (~100%) *No uplift needed*	Realistic possibility (20–70%) *Some uplift needed*
TA5	Almost certain (90-100%) *No uplift needed*	Almost certain (~100%) *No uplift needed*	Almost certain (90-100%) *No uplift needed*

3. How *willing* are threat actors to release data damaging worms?

In this section, we discuss how willing different threat actors might be to release data damaging worms *if* they were able to do so. We first analyse the motivations for past worm attacks (Section 3.1), and then discuss the incentives and disincentives for releasing data damaging worms (Section 3.2). We then estimate the probability that different threat actors would be willing to launch a data damaging worm (Section 3.3).

3.1. Motivations for historical worm attacks

In this section, our aim is to estimate the probability that, assuming they were able to do so, different groups of threat actors would release data damaging worms that could cause severe economic harm (>$1B in damage). One way to make progress on that question is to analyse the motivations for past significant worm attacks in the Johansmeyer (2024) dataset. Before we review the motivations for past attacks, it is important to make some conceptual clarifications.

Distinguishing intended and foreseen harms

When analysing the motivations for these attacks, it is important to distinguish:

1. Their ultimate intended aims.
2. The side-effects foreseen by the attacker prior to the attack.
3. Their unintended and unforeseen side-effects (i.e. accidental harms).

For example, the ultimate intended aim of the WannaCry attack was to raise money for North Korea, but a foreseen side-effect of the attack was that it would cause severe economic harm.[45] The ultimate intended aim of the NotPetya attack was to damage Ukraine, but it caused unintended collateral damage to companies outside Ukraine, including companies in Russia. The severe economic harm caused by data damaging worms could therefore be the ultimate aim of the attack, a foreseen side-effect, or an unintended and unforeseen side-effect.

Because worms are difficult to control, many worm attacks have caused significant accidental damage. It is notable that a substantial fraction of the economic damage from NotPetya, perhaps the most economically damaging worm attack ever, may have been accidental.[46] Nonetheless, we think it is plausible that, if future AI does uplift threat actors to find elite exploits, most of the expected damages will come not from accidental harms, but from attacks where severe economic harm is intended or foreseen.

[45] Some people interpret foreseen side-effects of actions as intended effects; people's intuitions about the meaning of 'intention' differ (Wagner 2014). We stipulatively distinguish them here for conceptual clarity.

[46] We discuss this in more detail in section 4.

Categories of motivations for past attacks

Based on our analysis of past significant worm attacks, the intended aims of past attacks can be divided into the following categories:

1. **Financial**: attackers release worms to acquire money.
2. **Broad destruction**: attackers aim to cause widespread destruction (e.g. global or continental).
3. **Limited destruction**: attackers aim to cause damage to narrow targets, such as specific countries or companies.
4. **Other**: some attackers seem to have been motivated by curiosity or to prove hacking skill.

Different types of worms

As noted in section 1.3, past significant worm attacks have had different functionality. They have included data damaging worms, but also worms with no destructive payload, and worms that create botnets to launch targeted attacks or send spam. Although most past significant worm attacks have not been data damaging worms, they still provide useful information about the willingness of different threat actors to launch data damaging worms in the future. We think it is likely that, for some of these past attacks, if the threat actor were able to create a data damaging worm, they would have done so because, as discussed in section 1.3, data damaging worms seem especially well-suited to achieving these aims compared to the alternatives. We think it is plausible that the reason they did not use data damaging worms is lack of technical capacity. Therefore, by understanding the base rate of threat actors with given motivations, we can understand how likely threat actors would be to launch data damaging worms in the future, if AI enabled them to do so.

Summarising the motivations for past worm attacks

Table 3.1 below shows the perpetrator and motivation for significant worm attacks in the Johansmeyer (2024) dataset. See Appendix A.7 for more detailed discussion and sources.

Table 3.1. *Motivations for and perpetrators of past major worm attacks (1998-2023)*

Event	Year	Actor	Data damaging worm?	Ultimate intended aim	Description
Melissa	1999	TA1/2	n	Accidental/broad destruction?	Damage due to creation of large network traffic, possibly accidental
ILOVEYOU	2000	TA1/2	y	Broad destruction/to prove hacking skill?	Damaged and corrupted data on infected systems
Klez	2001	TA1/2?	n	?	Disabled antivirus software, caused damage by creating high network traffic
CodeRed	2001	?	n	Limited/broad destruction?	Defaced websites on affected servers, launched Denial of Service attacks against specific websites, and created high network traffic
Nimda	2001	?	n	?	Created high network traffic + gave admin privileges on infected systems
SirCam	2001	TA1/2?	y	?	Could share confidential docs during spread, and could delete all files on a system in certain conditions
SoBig	2003	TA1/2?	n	Financial?	Distributed spam, created a large amount of network traffic
SQL Slammer	2003	TA1/2?	n	?	No malicious payload, caused damage due to large network traffic
Swen	2003	?	n	?	Disabled antivirus and firewalls, and created high network traffic
Mimail	2003	?	n	Financial	Launched Denial of Service attacks against anti-spam orgs. Some variants stole credit card information
Yaha	2003	TA3	n	Limited destruction	Created botnet to launch Denial of Service attacks against Pakistani websites
MyDoom	2004	TA1/2?	n	Limited destruction/financial?	Created a botnet to launch denial of service attacks against specific companies.
Sasser	2004	TA1/2	n	Broad destruction/to prove hacking skill?	Created network traffic causing companies to take systems offline
StormWorm	2007	TA3	n	Financial	Charged a fee for denial of service attacks against anti-spam websites and security vendors
Conficker	2008	TA3	n	Financial	Created large botnet for cybercrime, but was never used due to concerns about criminal repercussions
WannaCry	2017	TA4	y	Financial	Ransomware worm to raise money for North Korea

NotPetya	2017	TA5	y	Limited destruction	Designed to limit damage to Ukraine by wiping data on infected systems

Notes. ‘?’ means that the intended aim of the attack is uncertain.

As Table 3.1 shows:

- **Financial motivations**: a quarter of attacks were clearly financially motivated, and a further 12% may have been financially motivated
- **Broad destruction**: a quarter may have been launched to cause broad destruction
- **Limited destruction**: 12% clearly aimed to cause limited destruction, and a further 12% may have aimed to cause limited destruction.
- **To prove hacking skill**: 12% of attacks may have been released to demonstrate the attacker(s)' hacking skill.
- **Unknown motivations**: 30%.

For the attacks with unknown motivations, it seems unlikely that they were financially motivated, as there is no obvious way they could have enabled the attackers to acquire money. So, it is reasonable to assume that these were either released to cause broad destruction or to prove hacking skill.

We think that worm attacks will generally pose less risk when attackers only intend or foresee *limited* damage, compared to other types of worm attack. It is plausible that attacks motivated by broad destruction pose the greatest risk, other things equal, as attackers are explicitly optimising to cause severe economic harm.

3.2. Incentives and disincentives to release data damaging worms

Data on the motivations for past worm attacks is sparse, so it is also useful to consider the incentives for and against releasing worm attacks that could cause severe economic harm.

3.2.1. Destructive motivations

In Table 3.2, we summarise the incentives for and against different threat actors releasing worms purely to cause destruction.

Table 3.2. *Incentives for and against releasing worms for destructive reasons*

	Incentives for	**Incentives against**
Non-state actors (TA1-3)	**Political ideology** seems to motivate many non-state attackers. **To demonstrate their hacking skill,** many attackers launch attacks which cause significant disruption.	**Criminal repercussions.** Attackers have often faced criminal repercussions for worm attacks. **Political repercussions.** Governments often sanction non-state actors for cyber attacks (TRMLabs 2024).
State actors (TA4-5)	**Destructive worms may be more attractive to rogue states** who are less concerned about political and diplomatic repercussions. **Destructive worms may be more attractive during active conflict.** Even though states	**Political repercussions for attackers.** Because worms can cause large amounts of damage, there is a risk of political and diplomatic repercussions. **Risk of blowback to attackers and their**

	usually lack the incentive to launch worm attacks, this may not be true during active conflict.	**allies.** Worms are difficult to control, and there is the risk that the worm also damages the state that launched the attack and its allies. **Most states rarely aim to cause as much damage to the world as possible via any means,** including cyberattacks. **Opportunity cost.** Launching destructive worms requires very high end cyber capabilities, which can be used for other offensive cyber operations, such as espionage. State espionage is also *de facto* tolerated under current international norms, whereas destructive cyberattacks are not. **Cyberweapons may be less cost-effective than other methods**. Even if states do wish to cause destruction, cyberweapons are often less cost-effective than e.g. kinetic attacks (Lin 2022; Bateman 2022)

It seems likely that at least some TA1/2 actors would release data damaging worms if they could. For many of these actors, political motivations, a desire to demonstrate technical skill, or simply to cause a large amount of damage can outweigh the potential criminal repercussions.

Historically, TA3 group actors have launched worm attacks for limited destruction, but there are no known cases of these actors using worms where broad destruction was intended or foreseen. This is likely in part because of the risk of criminal repercussions. However, actors include some terrorist organisations, which may be willing to release data damaging worms in the future irrespective of the criminal repercussions, though there are no known cases so far of terrorist groups launching significant data damaging worms.

Most TA4/5 actors usually lack the strategic incentive to launch worm attacks purely to cause broad global destruction, and there are no cases in the historical record. It may be that data damaging state worm attacks would be more likely during active military conflict. NotPetya aimed to cause limited destruction and was launched during active military conflict between Russia and Ukraine. However, Russia has not used any cyber worm weapons since (Bateman 2022a, p. 21) and we are not aware of any other state having done so. This may reflect a reduced willingness to use worms due to the potential backlash from their collateral effects on non-combatant countries (Bateman et al 2022b).[47]

As noted above, even if states do not intend to cause significant destruction, they may still have incentives to create worms that then unintentionally cause significant collateral damage.

[47] It could also be explained by a lack of technical capacity, or the opportunity cost of using elite exploits in worms rather than for espionage

3.2.2. Financial motivations

Financially motivated significant worm attacks have usually been launched by non-state actors – with North Korea's WannaCry the only exception to this rule. Table 3.3 summarises the incentives for and against using worms for financial reasons.

Table 3.3. *Incentives for and against releasing worms for financial reasons*

	Incentives for	Incentives against
Non-state actors (TA1-3)	**Worms allow larger reach than targeted approaches,** and so could gain more money, other things equal.	**International pressure.** Worms that spread to a large number of computers and cause a large amount of collateral damage are likely to bring greater criminal repercussions for the attacker, compared to targeted approaches, due to international diplomatic pressure. **Domestic pressure.** There is a risk that the worm could unintentionally infect systems in a state that usually tolerates the attacker. This would increase the risk that the harbouring state punishes the attacker. **Higher negotiation costs.** Trying to extract a ransom from a large number of actors hit by a worm, rather than a smaller number of higher value targets, involves higher negotiation costs for the attacker.[48]
State actors (TA4/5)	**Cybercrime may be attractive to states that are heavily sanctioned**. The most obvious example is North Korea, which has raised billions of dollars via cybercrime (CNN 2023), though notably little from WannaCry.	**Most governments do not need to use cybercrime to raise money**, as they can raise money via taxes. For most governments, cybercrime is not worth the political or diplomatic costs. **Worm attacks risk blowback to the attacker state or to their allies.** North Korea has limited internet-facing infrastructure, so this is less of a concern for them, but worms could still hit their allies, such as China.[49]

For non-state actors, there is a trade-off between the reach promised by worms (and therefore how damaging it is) and the risk of criminal repercussions. This may explain why there are no past cases in which non-state cybercrime groups have launched worms designed to infect and damage as many systems as possible. For example, the Conficker worm unintentionally spread so far that it drew significant attention from the global cybersecurity community and therefore became "too hot to use" by the Ukrainian cybercrime group that launched it (NYT 2019).[50]

[48] Ransomware attacks often involve extensive negotiation with the victim (The Economist, 2024).

[49] China provides substantial aid to North Korea and accounts for most of its trade (NKNews 2024).

[50] The Russian state rarely prosecutes cybercriminals (Maurer, 2018) but did arrest the DarkSide hacking group that caused the Colonial Pipeline attack after it caused a diplomatic incident with the US (Politico, 2022) (note this was not a worm attack). The BlackMatter ransomware group explicitly avoids targeting certain sectors for this reason (The Record 2021).

By far the most concerning financially motivated state actor is North Korea. Due to sanctions, they are especially willing to try to raise money via cybercrime and are less concerned about political or diplomatic repercussions (UN, 2024). WannaCry was apparently released by North Korea for financial reasons, but due to errors in the code and design of the bitcoin payment system, the bitcoin addresses only ever received $250,000 (Elliptic 2017), and it is unclear whether North Korea cashed out any of the funds (Carbonite 2017; Coburn et al 2019, p. 44).

WannaCry would have made more money if North Korea had fixed the errors in the WannaCry code. Nonetheless, it is notable that North Korea has apparently had much more success with targeted attacks (ElectricIQ 2020). For example, North Korea stole $81M in a cyber attack on a Bangladeshi Bank (DOJ 2018) and recently stole up to $1.5B from a cryptocurrency exchange (Guardian 2025), though it is unclear how much of the stolen funds they will be able to capture (Darknet Diaries 2020).

3.3. Summarising the willingness of threat actors to launch data damaging worm attack

In light of the discussion in this section, the table below summarises our willingness estimate for different threat actors. These are estimates of the probability that *at least one* threat actor in each threat actor class would be willing to launch a data damaging attack if they could. It is *not* the probability that *a randomly selected* threat actor in the threat actor class would be willing to launch an attack if they could.

In this table, for ease of analysis, we separate out OC3 terrorist and OC3 cybercrime groups, as they face fundamentally different incentives to launch worm attacks optimised to cause damage.

Table 3.4. Rationales for our current best guess for the annual probability that at least one threat actor in each subset of threat actors would launch a data damaging worm attack if they could[51]

Threat Actor	**Rationale**	**Willingness (90% CI)** Annual prob. *any* actor does attack *if* they could succeed	**= an attack every x years (90% CI)**
TA1	• It is difficult to know the skill level of perpetrators of past worm attacks - whether they were by TA1 or TA2 actors. From 1998-2005, there were 4-9 destructively motivated worm attacks by TA1/2 actors. This implies an attack every 0.7-1.5 years. Though the high end estimate may include worms for which *broad* destruction was intended or foreseen.	Extremely likely (98-100%)	One attack every 1 years

[51] Detailed calculations are available in the 'Expected damages AI uplift' tab of Worm damages

	● Survey results suggest that there are ~1M TA1 actors. It seems extremely likely that from such a large sample, at least one would be willing if they were able.		
TA2	● See TA1 considerations. ● Survey results suggest that there are 10K to 100K TA2 actors. It seems very likely that from such a large sample, at least one would be willing if they were able.	Highly likely (70-100%)	One attack every 1 to 1.5 years
TA3 cybercrime	● In RAND (2024), examples of TA3 actors include cybercrime groups, industrial espionage organisations, and terrorist groups. Cybercrime and industrial espionage seem extremely unlikely to release a worm that is optimised to cause damage, though there have been cases of narrower TA3 attacks. ● TA3 actors were very likely capable from 1998 to 2008, but there are no precedents of TA3 cybercrime groups launching worms optimised to cause damage or where broad destruction is foreseen	Very remote chance (0.5%-1%) *No incentive*	One attack every 100 to 200 years
TA3 terrorist	● TA3 terrorist groups might be willing. However, no major worm attacks have been launched by terrorist groups. For past attacks with an unknown perpetrator, terrorist groups would plausibly have claimed credit if they had launched them. ● From 1998-2008, TA3 actors could have released worms optimised to cause damage if they had wanted, since TA1/TA2 actors were able to do so up until 2004, and TA3 actors released several targeted worms up until 2008. Yet there were no such attacks. As an upper bound, this suggests that we should expect to see an TA3 worm attack at most every 10 years, or an annual risk of 10%. ● There are plausibly one to two orders of magnitude more TA1&TA2 actors than TA3 actors. If the distribution of motivations between these two sets of actors is the same, this implies a discount on TA1/2 risk of 10-100X, which implies an TA3 risk per year of roughly 2-17%. This may be biased high because the distribution of motivations might not be the same. So,the possibility of some or all members of a 10 person TA3 group sharing these motivations is more conjunctive than for TA1/2 actors.	Highly unlikely (2–5%) *Small subset may want to*	One attack every 20 to 50 years
TA4	● The most concerning actor is North Korea - limited risk of blowback and shown past willingness to launch such attacks. ● One interpretation of the WannaCry case is that North Korea would launch more worm attacks if they had elite exploits. ● Assuming that North Korea was able to launch major worm attacks from 1998 to 2009 (since TA1-3 actors could in that period).[52] There were no TA4 attacks in that period. This implies an upper bound risk per year of 8%. ● North Korea may now be less willing to launch worm attacks with wide reach due to the operational failure of WannaCry.	Highly unlikely (2–5%) *Small subset may want to*	One attack every 20 to 50 years

[52] North Korea's primary cyberattack unit opened in 1998 (New York Times (2015)).

TA5	● TA5 states would face significant blowback and diplomatic repercussions. ● Russia suffered significant damages from NotPetya, which may in part explain why they haven’t released worms since. ● No cases of TA5 broad destructive attacks, despite clear capability through 1998-2023. This suggests at most an annual risk of 4% per year. ● TA5 states seem to lack the strategic incentive outside wartime. ● Conditioning risk on war conditions on a rare event. Forecasting platforms suggest 0.4%-4% annual risk of war between TA5 actors. NotPetya is the only case of TA5 actors using destructive worms during war. Generously assuming risk of deployment conditional on war of 20%-50%, implies overall risk of 0.1%-2%.	Very remote chance (0.5%-1%) *No strategic incentive*	One attack every 100 to 200 years

4. Potential damages from data damaging worm attacks

So far we have reviewed how willing and able actors are of releasing data damaging worms. In this section, we review the evidence on the potential economic costs of data damaging worms, assuming that AI uplifted different threat actors to find elite exploits.

We first review which types of economic costs we try to quantify (Section 4.1). Next, we discuss and critique the evidence on the economic damages of past worm attacks, and argue that WannaCry and NotPetya caused economic damages of ~$1B and ~$10B respectively (Section 4.2). We then discuss the potential economic damage of data damaging worms using elite exploits if they were released today (Section 4.3). We argue that WannaCry and NotPetya could have been far worse with relatively modest changes in their code, but that the risk may be lower today due to improved cybersecurity.

4.1. Estimating economic damages from worm attacks is challenging

As discussed in Section 1, we only focus here on economic damages, and not other kinds of damage, such as geopolitical effects on the global balance of power. However, even economic damages are difficult to quantify. Following the White House’s CEA (2018)’s taxonomy, we include both *direct* costs to firms infected by the worm, and also *indirect* costs to suppliers, intermediate customers, and end consumers who were not infected by the worm. Both direct and indirect costs include lost revenue due to business interruption, reduced consumption, costs of cleaning up or replacing infected systems, legal payouts, and insurance costs.

Estimating the economic costs of worm attacks is challenging because it involves collecting data on a large number of affected parties across jurisdictions, and the available statistics are fragmentary (Anderson et al., 2019). Indirect costs are particularly difficult to measure as they are also influenced by other complex economic events, so isolating what was actually caused by a cyberattack is difficult. It is important to note that some types of economic cost often used in economic damage estimates do not always translate into deadweight social loss. For instance, if a firm loses revenue due to a cyberattack, demand might shift to a rival company, which could offset some of the social costs. The following sections try to take multiple independent approaches from which we can try to produce an overall estimate.

4.2. Historical data on the damages of past worm attacks is limited but WannaCry and NotPetya plausibly caused damages of $1B-$10B

4.2.1. Data on damages from past worm attacks is limited

There is limited data on the damages of past worm attacks. Johansmeyer (2024) collected various publicly reported damages from ‘significant’ cyber attacks (>$800M damage and >10-25 companies affected) from 1998 until today.

These data suggest that some worm attacks prior to 2005 caused large amounts of damage (e.g. $65B for the SoBig worm and $67B for MyDoom). However, Johansmeyer notes that the provenance of his data on cyberattack damages is poor, and most of these estimates in general seem unreliable (Johansmeyer nd; see also Cobos & Cakir, 2024). The main problems are as follows. (For more discussion, see Appendix A.8).

1. There is no methodological information or calculations for any of the estimates. Many of the early damage estimates come from consultancies who do not share their methods, and some argue that they have incentives to produce inflated estimates (Leyden, 2002; RTI 2006).
2. Damage estimates for the same event vary by orders of magnitude. Johansmeyer (2023, fn. 1) notes that where there are multiple sources, he chooses the higher estimate. (This is because Johansmeyer’s aim is in part to show that even on the high estimates, cyber risk is lower than many people argue.)
    a. For instance, one of Johansmeyer’s sources for damage estimates is the November 2003 House Committee Hearing on computer viruses. In that hearing, different speakers and sources gave damage estimates for SoBig ranging from $500M to $30B.
    b. One source used in the dataset reports that the Melissa worm caused >$1B in damage (in 2012 dollars), providing no calculations or source (Beattie 2012), but in the plea agreement, the Melissa perpetrator admitted to causing “over $80M” in damage (The Register 2001).

It generally seems difficult to make progress on estimating the damages of earlier worms, as there is a lack of good evidence. We focus on damages of the two most recent worms which are the closest analogue for the threat model considered here: WannaCry and NotPetya.

4.2.2. NotPetya damages were plausibly $3B-$10B

After searching the academic literature, we found only one study (Crosignani et al (2023))[53] which has verifiable calculations of the overall social damages of WannaCry or NotPetya. We

[53] An open access working paper version of Crosignani et al (2023) is available here.

also develop our own less robust estimate of NotPetya based on claims made by Ukrainian government officials.

Crosignani et al (2023)

The methodology of Crosignani et al (2023) is as follows:

- **Direct damages**: They constructed a list of eight (non-Ukrainian) firms that were directly hit by NotPetya and collated their reported losses, including lost or delayed revenue, and remediation costs.
- **Indirect costs**: They then identified companies in the upstream and downstream supply chain for the affected companies and compared their financial performance to similar unaffected companies.

They estimate that the direct damages were $1.8B, and downstream supply chains suffered “conservative” damages of a $7.3B drop in revenue compared to the counterfactual. This implies a total of $9.1B in damages in 2017 dollars, which is ~$11B in 2023 dollars.

This result is potentially biased in either direction. It could be biased *high* because lost revenue for specific companies in a particular period may not reflect true social loss. Rival firms might gain revenue, and sales might be higher in later periods. It is difficult to estimate how large this effect might be.

The estimate could be biased *low* because the damage estimates do not include governments or Ukrainian companies, even though Ukraine accounted for 75% of total infections. Naively, this suggests that the costs might be biased high by a factor of four,[54] but Ukrainian income per person was much lower than other affected countries, so countries outside Ukraine likely suffered higher economic costs per NotPetya infection.[55]

Overall, it is difficult to know how these two effects balance out, but we think Crosignani et al (2023) provides some evidence that **damages of ~$10B are roughly plausible**. We discuss the study in more detail in Appendix A.9.

Inferring global damages from claims about Ukrainian damages

Another (less reliable) approach to estimate NotPetya damages involves the following steps

- The Ukrainian Finance Minister claimed that NotPetya cost 0.5% of Ukrainian GDP (Hromadske 2017), or $560M in 2023 dollars.
- Since there were ~500K infections in Ukraine (Maschmeyer 2021), this implies a cost per infection of ~$1.1K.

[54] Because assuming losses are proportional across countries, Ukraine accounts for three quarters of losses, which are not reflected in the Crosignani et al estimate.

[55] However, for the same reason, the costs to *social welfare* in Ukraine may have been higher because economic losses are worth more to Ukrainians than to richer people. This illustrates a broader problem with expressing social loss in terms of dollars. We discuss this in Appendix x.

- We then adjust the cost per infection in different countries based on how their GDP per capita compares to Ukraine: for example, because German GDP per capita was 17X Ukraine's in 2017, we assume that the cost per infection was 17X. We make this adjustment for all affected countries.
- Using a rough estimate of the number of infections in each country, we can then infer the cost per country and aggregate the total cost. This implies **total damages of ~$2.6B**.

Full calculations are in the 'NotPetya & WannaCry damages' tab of Worm damages.

This approach avoids some of the problems with Crosignani et al (2023) outlined above and discussed in more depth in Appendix A.9, but rests on an estimate from the Ukrainian Finance Minister which cannot be verified and seems unreliable. Indeed, in the same article quoting the Ukrainian Finance Minister, the Ukrainian head of cyber police stated that "These are all rather arbitrary calculations… Now we can only estimate the cost of the disabled computers, and somehow calculate the lost profits from the non-working services. But in reality, we still don't know what exactly the virus did" (Hromadske 2017).

Overall, we put more weight on the adjusted Crosignani et al estimate. This implies **damages on the order of ~$10B.** This implies a cost per infection of ~$17K.

4.2.2. WannaCry damages were plausibly $1B-$4B

We have not found any systematic and rigorous attempts to quantify the total damages of the WannaCry attack.[56]

Inferring WannaCry damages from NotPetya damages

One method involves inferring WannaCry damages from the NotPetya damage estimates outlined above. Although data is limited, it is reasonable to assume that the cost per infection from WannaCry was lower than for NotPetya, as the encryption method used by WannaCry was breakable in some cases and did less damage to infected systems.[57]

[56] There are estimates of the costs of WannaCry on the NHS. Ghafur et al (2019) estimate total costs of £5.9m, while DHSC 2018 estimate costs of £92m, which is 0.01% to 0.1% of the total NHS England budget. If we assume costs to the NHS were proportional to costs on the global economy, this implies costs of $4.4bn to $68bn. Calculations are in the 'NotPetya & WannaCry damages' tab of Worm damages.

[57] The NotPetya and WannaCry payloads each had similar functionality, but NotPetya caused more damage because it wiped the whole disk on an infected system, making the OS and all system files inaccessible (Malwarebytes 2017), whereas WannaCry focused on data files and some backup files (Computable nd; Forbes 2017). NotPetya deployed a faux ransomware message, but it was not possible to decrypt affected files (Kaspersky 2017). WannaCry encrypted files and decryption was not possible without access to a decrypted private key from the attackers (Sophos 2017), though decryption was possible in some limited cases (Bank Infosecurity 2017). Even if victims paid the ransom, there was no way of associating the payment with a specific computer (Kaspersky nd), and very few victims actually paid the ransom (QZ 2017).

Therefore, to set an upper bound on the costs of WannaCry, we can assume that the cost per infection for both attacks was the same. As Table 4.1 shows, this implies **upper bound damages of ~$4B**.

Table 4.1. Upper bound: Inferring WannaCry damages from NotPetya damages

Variable	Value	Source
NotPetya damages	$10B	Crosignani et al (2023)
Total NotPetya infections	664K	Inferred from Maschmeyer 2021, p. 81[58]
NotPetya cost per infection	$17K	Calc
WannaCry infections	230K	(Symantec 2018)
Implied WannaCry damages (2024 $)	**$4B**	Calc

Inferring from damages to the English NHS

There are more rigorous published estimates of the costs of WannaCry for the English National Health Service (Ghafur et al (2019)), one of the most high-profile WannaCry victims (BBC, 2017). Ghafur et al (2019) only estimates costs to secondary care (hospitals, specialist clinics etc), not to primary care (GPs, dentists, etc), and finds damages across the English NHS of £6M. We therefore have to infer total costs to the whole NHS, and then infer total global damages from that. Our best guess estimate implies **damages of around $1B**,[59] but this involves various highly subjective and uncertain assumptions, so we do not have much confidence in this estimate.

Though we have less confidence in these estimates than the NotPetya damage estimates, we think **a credible range for WannaCry damages is roughly $1B-$4B**.

4.3. With modest changes, WannaCry and NotPetya could have done far more damage

4.3.1 WannaCry and NotPetya could have done far more damage

Although WannaCry and NotPetya caused damages of $1B-$10B, with modest changes they could have done far more damage.

[58] This is inferred from a claim in Maschmeyer 2021. For NotPetya, there were 500K infections in Ukraine alone (Maschmeyer 2021). 75% of infections were in Ukraine (ESET 2017), which implies ~670K infections globally. Maschmeyer's estimate is difficult to verify. Maschmeyer notes that "This estimate by an anonymous expert at a leading cybersecurity vendor in Ukraine is based on the number of compromises he observed personally while involved in mitigation efforts at multiple large enterprises in Ukraine."

[59] Full calculations are in the 'NotPetya & WannaCry damages' tab of Worm damages.

WannaCry and NotPetya only infected hundreds of thousands of systems,[60] whereas earlier worms had much greater reach.[61] It seems clear that the potential damage of both WannaCry and NotPetya could have been far greater if the attackers had not made various mistakes or had been optimising to do as much damage as possible.

- **WannaCry included a 'kill-switch', which a researcher was able to find and activate 7 hours into the attack.** WannaCry was designed to check whether a certain gibberish URL led to a live web page, and if it did, the malware stopped further encryption on affected machines, and the spread of the worm was slowed significantly (Kryptos Logic (2017); Malwaretech 2017; Cisco 2017). A computer security researcher registered the domain for a small fee and effectively stopped further damage from the attack (WIRED, 2017; Kryptos Logic (2017)). Absent this unforced error it is plausible the attack could have infected and damaged far more systems.
- **NotPetya was designed to target damage to Ukrainian systems**.[62] According to a Ukrainian government official, 10% of all computers in Ukraine were wiped by NotPetya, and half of these were not recoverable (Hromadske 2017). So, the damage could have been far greater if the worm had been designed to hit all countries. The most plausible way this could have worked would have been if NotPetya had used EternalBlue to spread between networks (like WannaCry) rather than only within networks. This would involve only relatively modest changes to the NotPetya code which seem within reach for the Russian actors that launched the attack.
- **Both WannaCry and NotPetya exploited a vulnerability for which a patch was available 1-2 months prior to the attack.** Many potential victims would have patched their systems in the two month gap between the release of the patches and the release of the worms.

This is important for the AI-enabled data damaging worm threat model. Firstly, future AI could reduce the risk of errors in worm code, which would otherwise limit potential damage. Second, as discussed in section 3, AI could uplift actors (such as TA1/2 lone wolves or TA3 terrorists) who aim to cause broad destruction rather than more limited damage. Third, AI could enable threat actors to find elite exploits of zero-day vulnerabilities with no available patch.

[60] For NotPetya, according to one source, there were 500K infections in Ukraine alone (Maschmeyer 2021). 75% of infections were in Ukraine (ESET 2017), which implies ~670K infections globally. For WannaCry, there were >230K infections within 8 hours (Symantec 2018).

[61] ILOVEYOU infected 50M within 10 days (Forbes 2020), and SoBig infected tens of millions of systems (Esecurityplanet 2004).

[62] NotPetya nonetheless spread to other countries. For instance, NotPetya spread to Maersk's networks because M.E.Doc was installed on a single computer in Maersk's Ukraine operation (WIRED 2018).

4.3.2. Different methods imply that if worst-case versions of WannaCry and NotPetya were released in 2017, they would have caused tens of billions to $100B in damage

We outline three methods to estimate potential damages if worst-case versions of WannaCry or NotPetya were released in 2017. All of these methods involve naively extrapolating from the estimated damages of WannaCry and NotPetya to worst-case versions that spread globally. After describing these three naive models, we will discuss their limitations.

It is important to stress that these estimates are about damages if worst-case versions of WannaCry and NotPetya *were released in 2017*. As we discuss in section 4.3.3, this does not necessarily imply that if those worms were released *today* that they would do comparable damage because cybersecurity has improved since 2017.

Method 1: A simple and unreliable model of "WannaCry without the kill switch"

If WannaCry had not included a kill switch, it seems plausible that it could have done far greater damage. WannaCry infected ~230,000 systems within around 7 hours. Kryptos Logic (2017) claims that kill switch directly prevented 14M to 16M infections and reinfections – and that "well into the tens of millions or higher" computers could have been reached if the kill switch had not been activated. This suggests that, if WannaCry had not included the kill switch, infections could have been around 50-200X higher.

If systems were also unpatched, the reach could have been even greater. At the time, the total pool of systems that could have been damaged by WannaCry, provided they were unpatched, was likely around 400M at the time of the attack.[63] So, if WannaCry did not include the kill switch and systems were unpatched, infections could potentially have been up to 1,700X higher.

Table 4.2 shows the number of infections on different scenarios, on the following assumptions

- Constant doubling time: Chernikova et al., 2023 suggests that WannaCry was spreading exponentially before the kill switch was activated. The doubling time of WannaCry before the kill switch was activated was around 1.2 hours.[64]
- There was no kill switch.
- All systems were unpatched, i.e. perhaps the malware exploits zero-day vulnerabilities.

[63] As of 2016, according to one estimate, there were ~1.5 billion active Windows users (Thurrott 2016). However, the pool of potentially vulnerable systems was much smaller than this. Microsoft claimed that no known Windows 10 users were infected by WannaCry (Microsoft (2017)). Coburn et al (2019), p. 44 claimed that at the time of the attack, there were 400 million actively used Windows computers running version 8 or an earlier operating system. This is therefore a more plausible upper bound on potential infections.

[64] In one lab experiment, WannaCry did not spread exponentially, though this may be because the lab test used a network with only 50 hosts (Nguyen et al 2024, p. 5). Given how fast WannaCry in fact spread during the actual attack and its mechanism of propagation, it seems like it must have spread exponentially. So, we do not put much weight on this point.

The final column shows damages on different scenarios, assuming that damages scale with infections.

Table 4.2. Simple model of damages of a worst-case version of WannaCry[65]

		Infections	Damages
WannaCry Actual (after 7 hours)		230K	$2B
Extra hours spread	2	737K	$6B
	4	2M	$21B
	6	8M	$66B
	8	24M	$210B
	10	78M	$674B
	12	248M	$2.2T
	13	444M	$3.9T

As we discuss below, **we think the damages implied by this simple model are too high.**

Method 2: Inferring global damages from proportionate damages to Ukrainian GDP

As discussed above, the Ukrainian Finance Minister claimed that NotPetya reduced Ukrainian GDP by 0.5%. Table 4.3 outlines global costs if NotPetya were designed to spread globally and did proportionate damage to the world economy.

Table 4.3. Inferring potential global damages from one unreliable estimate of damages to Ukrainian GDP

Variable	Value	Source
Proportionate costs to Ukraine GDP	0.5%	Ukraine Finance Minister (Hromadske 2017)
World GDP in 2017 (2023$)	$81.7T	World Bank
Implied costs to world GDP	**$408B**	Calc

We do not have much confidence in this estimate because, as noted in section 4.2.2, it is based on an estimate of the cost of NotPetya to Ukraine which is difficult to verify and seems unreliable.

Method 3: Inferring global damages from one estimate of the fraction of computers destroyed in Ukraine

A Ukrainian official claimed that NotPetya destroyed 5% of all computers in Ukraine. Table 4.4 infers global damages for a worst-case version of NotPetya from this, assuming a given cost to replace a damaged computer.

[65] Note that due to rounding, infections and damages may appear not to scale in accordance with the assumptions of our model. Note that the total pool of vulnerable systems was ~400M (assuming no systems were patched), which is why the table only considers spread of up to 13 hours.

Table 4.4. Inferring potential global damages from one unreliable estimate of the fraction of computers destroyed by NotPetya

Variable	Value	Source
% of Ukrainian computer destroyed by NotPetya	5%	Ukrainian official (Hromadske 2017)
Number of computers, global	~2B	SCMO 2019
Total computers destroyed	100M	Calc
Cost to replace computer	$500	Assumption
Implied costs to world GDP	**$25B**	**Calc**

This method does not include indirect costs and so is biased low in that respect.

Problems with naive extrapolation models

Defender response might reduce total infections

These 'naive extrapolation models' are plausibly biased high. The longer the attack lasts, the more time defenders would have to respond by taking systems offline and patching systems. This limits the potential spread of worst-case versions of WannaCry and NotPetya. Moreover, the spread of the worm would likely have followed an s-curve, rather than having a constant doubling time across the whole attack.[66] The number of potential hosts that could be infected by each node would decline in the later stages of the attack as more potential nodes are already infected. This in turn gives more time for victims to respond. Spread would likely have slowed down substantially after a third to a half of vulnerable systems were infected, giving defenders more time to respond.

However, attackers could reduce the scope for reactive response by delaying the execution of the payload until a large number of systems are infected,[67] though this would also make writing the worm harder.

It is difficult to judge how this affects the plausibility of the naive extrapolation models, but we think it makes damages of >$200B less plausible.

[66] There are a range of studies exploring infection dynamics for worm malware, which use epidemiological models to model propagation dynamics. These studies typically use epidemiological models which imply s-curve spread over the full course of an outbreak (Chernikova et al., 2023; Martinez et al 2021; Baksi and Upadhyaya 2020).

[67] The Stuxnet payload only activated when it detected the presence of Siemens control systems used in the Natanz nuclear centrifuge plant (Ars Technica 2011). EternalRocks only beaconed out to its command and control infrastructure after a delay (BleepingComputer 2017). For other malware, the payload could be programmed to execute at a specific date or time, or after communication with attackers' command and control.

Damages might not scale with infections

These models assume that damages scale in proportion to infections, but this might not be accurate.

- Once (e.g.) 10% of a company's systems are infected, they already have to shut down operations, so the remaining 90% of infections may be much less damaging.
- Cleaning (e.g.) 100M devices may not be 100x as expensive as cleaning 1M systems because you can reuse the same methods in both cases.

We are unsure how much we should adjust the estimate above in light of this point, but our subjective guess is that this makes damages of >$100B less plausible.

Table 4.5 summarises the results of the naive extrapolation models and our evaluation of them

Table 4.5. *Summaries of different estimates of the social costs of worst-case versions of Wannacry*

Method	Damages	Evaluation
1. Extrapolation from cost per infection	$6B-$3T for 2-13 hours extra of spread	Upper end of range biased high.
2. Infer from cost to Ukrainian GDP	$408B	Uncertain. Based on unreliable assumptions.
3. Cost to replace destroyed computers	$25B	Biased low. Only includes the cost of replacing computers.

Method 4: Lloyds Basche scenarios imply damages of $85B-$193B from worst-case worms

Thus far, we have constructed our own simple models of worst-case versions of WannaCry and NotPetya. In this section, we discuss estimates constructed by the insurer Lloyds of extreme data damaging worm scenarios.

Lloyd's (2019) outlines a set of worm scenarios similar to a worst-case version of WannaCry, which they call the 'Basche scenarios'. Lloyd's describes the scenarios as unlikely but plausible (p. 10).[68] They outline three scenarios, S1, S2 and X1. In all scenarios, the malware is sent to each company via a phishing email, rather than directly via the EternalBlue exploit as in the actual WannaCry attack. Once a single employee downloads the file, the worm spreads to other systems on the company's network, and then forwards the malicious email to all contacts within infected devices' address books (p. 13).

[68] In the Basche scenario, the relevant actor is a cybercrime syndicate. However, on base rates, it seems unlikely that such groups could find the requisite exploits (as discussed in Section 2), or would use a worm rather than more targeted attacks to make money.

Table 4.6 outlines the assumptions of the three scenarios, and the implied direct and indirect economic damages:

Table 4.6. *An overview of the Lloyds Basche scenarios*

Variable	S1 Scenario	S2 Scenario	X1 Scenario
Malware targets operating systems running on what fraction of global devices (p. 19)	43%	97%	97%
Fraction of companies infected in different sectors (Table 1)	1-9%	2-16%	3-21%
Number of infected companies (Table 6)	250,000	501,000	613,000
Fraction of systems infected for the median affected company (Table 4)	~15%	~15%	~15%
Payload (p. 21)	Ransomware	Ransomware	Ransomware and wiper which deletes backup files.
Number of encrypted computers	~30M (p. 13)	~60M? (inferring from S1 scenario and # of infected companies in S2)	~73M? (inferring from S1 scenario and # of infected companies in X1)
Number of computers, global, 2019 (SCMO 2019)	~2B		
Fraction of total global computers infected (calc)	1.5%	3%	3.6%
Total direct economic loss (Table 6)	$59B	$110B	$133B
Productivity and consumption loss	$50B	$93B	$112B
Clean-up loss	$8B	$15B	$18B
Cyber extortion loss	$1B	$2B	$2B
Total indirect economic loss (Table 6)	$26B	$49B	$60B
Total global economic loss (Table 6)	$85B	$159B	$193B
Total damages per infected system (calc)	$2.8K	$2.6K	$2.6K

Estimated total infections in the different scenarios depends on:

1. **Risk of initial infection**: the number of companies infected by the attack.
   a. This is based on a 'Sectoral Vulnerability Score' developed by Lloyd's, with input from subject-matter experts. The score is determined by (1) the sectors' historical

susceptibility to ransomware delivered by phishing, and (2) the defensive capabilities of those sectors (p. 19).
    b. Lloyd's constructed an 'industry exposure dataset' to estimate the size and sector of different companies across the global economy (p. 28). This can be combined with the risk of initial infection to estimate the total number of companies infected.
2. **Spread within companies**: The fraction of computers in a company that are infected once the worm has gained access to a company's systems.
    a. This is also based on the Sectoral Vulnerability Score. Lloyd's (2019) say they "completed extensive research on worm propagation and found that worms have an upper limit of propagation within internal networks. After consulting with subject matter experts, the most accurate upper bound for infection was decided to be 40%+ to remain technically feasible" (p. 21), and most sectors have infection rates of 10-20% (Table 4).

The total number of infected firms is then obtained by summing up the expected infections across all types of companies.

Damages are determined by:

1. **Direct damages**
    a. Clean-up costs for each device ($350 per device) (p. 15).
    b. Lost revenue, which is determined by their estimates of the severity and duration of business interruption (p. 71), though we are unsure what these estimates are based on.
    c. Ransom fees ($700 per device) though only 4% of devices are decrypted by paying the ransom (p. 72), so this does not contribute much to overall damages.
2. **Indirect damages** are calculated using a 'contagion multiplier' for each sector. "The multiplier calculates the relative indirect revenue loss as a proportion of a sector's direct loss. The value of the multiplier was calculated by employing an input-output approach to estimate the relative indirect shock in inter and intra-sectoral trade globally using the World Input-Output Table" (p. 29). However, the report does not provide further details on how the multiplier was calculated.

Three points are notable about these estimates.

1. **Of the three scenarios, the S1 scenario is most similar to the worst-case WannaCry.** In the S1 scenario, the malware targets an operating system running on 43% of computers, whereas WannaCry was effective against ~20% of operating systems.[69]

[69] WannaCry was only effective against Windows systems running Windows 8 or earlier, provided they were unpatched. Coburn et al (2019), p. 44 claimed that at the time of the attack, there were 400 million actively used Windows computers running version 8 or an earlier operating system. Assuming ~2B total operating systems on computers globally, this implies the worst-case version of WannaCry was effective against (400M/2B=) 20% of computers.

2. **Substantially lower cost per infection than naive extrapolation models.** Lloyd's estimated cost per infection for the different scenarios (~$2.6K) is substantially lower than our own estimates for WannaCry ($4K-$17K) and NotPetya (~$17K). Although some parts of Lloyd's calculations are opaque, this may provide some evidence that our estimates of cost per infection in the naive extrapolation models of worst-case WannaCry and NotPetya damages are too high.
3. **Tens of millions, rather than hundreds of millions, of systems are infected.** In all scenarios, the number of infected systems is well below the total number of vulnerable computers: tens of millions of systems out of a total population of ~2B computers (~2-4%) are infected in the scenarios. As discussed above, this is mostly determined by Lloyd's estimates of potential spread between and within companies, which is determined by their own research in consultation with subject-matter experts. Although the report does not provide much information on the evidence underlying their estimates, this arguably provides some evidence that even in a worst-case scenario, it is only plausible that tens of millions of devices, or a small fraction of the population of vulnerable devices, would be infected. This may suggest that scenarios in which hundreds of millions of devices are infected, as in the naive WannaCry extrapolation in Table 4.2, are less plausible.
    a. However, the Basche scenarios deliver malware to different companies via phishing emails, whereas WannaCry delivered the EternalBlue exploit directly without requiring any user interaction. A worm that uses a zero-click exploit may be able to spread more widely. Indeed, in the WannaCry and NotPetya attacks, for some companies, far more than 40% of systems were infected.[70]

Overall judgement on damages from worst-case versions of WannaCry and NotPetya

We constructed three models that naively extrapolate damages for worst-case versions of WannaCry and NotPetya. Two of these models suggest that in the worst-case, these could have done hundreds of billions of dollars of damage, if not more. However, there are several reasons to think that damages in the hundreds of billions of dollars are too high.

The Lloyd's scenarios suggest that even in the worst-case, damages could approach $100B, but not much higher.

[70]

- 90% of a Ukrainian bank's computers were infected by NotPetya (Greenberg, *Sandworm* (2019), p. 180)
- 70% of Ukrainian Post Office computers were infected, despite efforts to shut down those systems after the attack was discovered (Greenberg, *Sandworm* (2019), p. 187).
- In a group of Ukrainian hospitals, "virtually all" Windows systems were encrypted (Greenberg, *Sandworm* (2019), p. 188).
- NotPetya came close to wiping all of Maersk's data. A backup domain controller in Ghana had been spared because there happened to have been a blackout there shortly before NotPetya hit (Greenberg, *Sandworm* (2019), p. 194).
- All Windows systems in an American hospital network were infected (Greenberg, *Sandworm* (2019), p. 201).
- 90% of Telefonica's systems were infected by WannaCry (Telecom Review 2017).

4.3.3. Potential damages from worms today

We have argued that with modest changes, WannaCry and NotPetya could plausibly have done ~$100B in damage *in 2017*. However, it is a further question whether a worm using an elite exploit could do comparable damage *today*. The most important reason that damages might be lower than this now is that modern cybersecurity systems would prevent substantial spread or damage.

Even if a worm exploits zero-day vulnerabilities or systems are unpatched, modern cybersecurity systems might be able to prevent a worm from spreading or deploying its payload. Only users using Windows 8 or earlier were vulnerable to WannaCry, and the majority of infections were for Windows 7 users (Bank Infosecurity 2017). Microsoft claimed that no known Windows 10 users were compromised by WannaCry (Microsoft 2017). Windows 10 protected against WannaCry due to more effective security features, including exploit mitigations, kernel protection, and machine learning-based antivirus and endpoint detection and protection (Microsoft 2018).

Microsoft also claimed that Windows 10 “either fully prevented or mitigated” the NotPetya malware (Microsoft 2018; Microsoft 2018), and one source reports that Windows 10 S blocked the attack by default (Bank Infosecurity 2017).

Cybersecurity has improved since 2017, so now it would be even harder for worms to infect and damage systems. Modern versions of firewalls, intrusion detection systems and endpoint detection and response systems can use machine learning and other techniques to detect malware on the basis of its network or endpoint activity patterns, rather than on the basis of known code signatures (Mauri and Damiani 2025; Karantzas and Patsakis 2021; Applebaum et al 2021). A single device scanning the network, or sending large numbers of packets or messages to other devices or networks might be noticed by these systems, helping to prevent significant spread. So, these systems might limit the potential reach of future worm attacks, even if they use 0-days.

For worms to overcome these security features today, they would have to be more complex than WannaCry or NotPetya, which increases the capability barrier that AI would have to help overcome. This consideration suggests that the damages from worst-case versions of WannaCry and NotPetya are at the higher end of what is feasible today. Therefore, we use these worst-case damages of ~$100B as a rough upper bound on potential damages. For a lower bound we use damages of $10B.

It is difficult to know how effective these systems might be against future worms without having concrete details on how such worms might be designed. This in turn depends on knowledge of the frontier of cyberoffence capabilities, which is difficult to gain without access to classified information.

Note that the discussion here is about *existing* cyberdefence systems. In the next section, we discuss how *future* AI vulnerability discovery capabilities might improve defence and reduce the potential costs of worm attacks.

5. Offence-defence balance

AI models that discover vulnerabilities and develop exploits are *dual use* in that they could be used by both attackers and defenders. In this section, we discuss how AI uplift to defenders might affect expected damages from worm attacks.

It is difficult to assess the effects of AI on the risk of worm attacks over time because offence-defence balance depends on a range of uncertain parameters (Lohn and Jackson (2022); Garfinkel and Dafoe (2019); Lohn 2025), and AI benefits attackers in some respects and benefits defenders in others. Table 5.1 outlines how AI relates to different determinants of the risk of worm attacks over time.

Table 5.1. How AI might affect different determinants of the risk of worm attacks[71]

Determinant of risk of worms	**Favours offence of defence?**	**Discussion**
Vulnerability discovery	**Unclear**	AI capability can be used by attackers and defenders.
Correlation of vulnerabilities found by attackers and defenders	**Unclear**	If vulnerabilities found by attackers and defenders are correlated, this favours defence. There are some reasons to think that automation will lead to greater correlation (Garfinkel and Dafoe (2019), p. 263). But correlation may be lower due to experimentation in prompting and scaffolding and use of different AI models.
Exploit development	**Favours offence**	This allows attackers to exploit faster. AI exploit development capability may be correlated with general AI coding abilities.
Patch development	**Favours defence**	Allows defenders to patch faster. This capability may also be correlated with general AI coding abilities. So, exploit development and patch development capabilities may be correlated.
Patch deployment	**Favours offence in short-term (<3m)**	Currently a key lag on defence. It seems harder for AI to affect patch *deployment* than e.g. exploit or patch development. Policy decisions have a greater impact on patch deployment rates.
Defenders can repair vulnerabilities prior to software release	**Favours defence in longer-term**	Defenders have a natural advantage in one respect in that they can use AI to repair vulnerabilities in their software *prior to the release of the software*. As new software replaces old software, the risk of cyberattacks declines. Apple and Google release a new OS around once a year, while releases one major feature update per year, and a major new OS every three years (Android 2025; Apple 2024; Microsoft 2025; Microsoft nd). So, plausibly defenders would start to benefit after around 6-12 months after the release of AI tools, as new AI-tested software replaces old software.
Warning shots	**Favours defence in longer-term**	Major cyberattacks serve as a warning shot which encourages defenders to improve cybersecurity. This broad dynamic seems to have occurred for worm attacks, which were high prior to 2005 but then declined once cybersecurity improved after numerous major worm attacks (see Section 1).

[71] The assessment here is based on Lohn and Jackson (2022), Garfinkel and Dafoe (2019), Lohn 2025, and the author's own judgement.

Table 5.1 suggests that the effect of AI on the risk of worm attacks will likely vary over time.

In light of the considerations above, it seems plausible that AI vulnerability discovery and exploit development capabilities will increase the risk of worms in the short-term because AI will improve (a) *vulnerability discovery* for both attackers and defenders, (b) *exploit development* for attackers, and (c) *patch development* for defenders, but there is less scope for AI to improve *patch deployment* rates. This suggests that AI uplift to factors (a), (b) and (c) will increase risk in the gap in which users deploy patches.

Data from 2008-14 suggests that half of users *deploy* patches after around 100 days (Lohn and Jackson 2022, p. 7).[72] We lack good data on patch deployment rates today, but we think it is plausible that today, for widely used software, 90% of patches would be deployed 0.5-3 months after patch release.[73]

So, in the absence of an active policy decision to improve patch deployment rates, we think that AI uplift for vulnerability discovery and exploit development would increase risk in the short-term (within 2 weeks to 3 months), but would decrease risk at some point after 3-12 months.

It is important to note that the conclusion one reaches about offence-defence balance in this domain does not settle the question of how AI might affect offence-defence balance in other areas of cyberspace. Offence-defence balance is a property of relationships between particular defenders and attackers, not of cyberspace in general (Slayton (2017), p. 74).[74]

For more discussion of offence-defence balance see Appendix A.11.

[72] Their source for this is Nappa et al (2015), Table III, which finds that 50% of users deploy patches after 15 to 268 days, and 90% after 129 to 799 days, depending on the software.

[73] We have not found more comprehensive recent data on patching rates for widely used software. In the 2021 Microsoft Exchange attacks, which, as discussed in Appendix A.7, may have involved elite exploits, 80% of servers were patched 10 days after patch release (Microsoft 2021), and >92% were patched 23 days after patch release (Microsoft 2021). We think these fast patch deployment rates likely reflect a broader improvement in patch deployment rates, but also are plausibly faster than average because the Exchange vulnerabilities were being actively exploited. Microsoft (2020), p. 23 suggests that 90% of patches are deployed after a week for Windows 10 users. Microsoft (2021) states that 80-90% of Windows patches are deployed after one month. We have been unable to find more recent data on patch rates for other widely used software. Cobalt (2025) claimed that the median time to resolve serious flaws found in pentests declined from 112 days in 2017 to 37 days in 2025, though this is different to a representative sample of patching rates across vulnerabilities. Various other metrics suggest that cybersecurity is improving over time (Healey and Jain (2025)).

[74] For example, the offence-defence balance for Chinese espionage against US government targets is different to the offence-defence balance for ransomware attacks against hospitals.

6. Results

Our overall aim in this report has been to estimate the expected social damages from data damaging worms *if* AI enables different threat actors to find elite exploits (including both finding vulnerabilities and writing exploits of them). As discussed in section 1, we constructed a simple model which breaks the estimate down into threat actor *capability* to create data damaging worms, threat actor *willingness* to launch data damaging worms, and *potential damages* if such worms were launched. In sections 2-4, we reviewed the evidence on these parameters and estimated probability distributions for each. We can now bring these estimates together to estimate the expected harm if AI gains elite exploit capabilities.

6.1. Scenario: AI uplift absent defensive benefits

As discussed in section 1, we want to estimate the *counterfactual* impact of AI capabilities. So, we estimate damages (1) on a 'Baseline' (no AI uplift) scenario, (2) assuming that AI uplifts different threat actors to find elite exploits.

The estimate in the Baseline column is given by the following equation:

> Expected damages from AI uplift to Threat Actor$_i$ = Severity$_i$ * Capability$_i$ * Willingness$_i$

The formula for marginal expected damages from AI uplift is, assuming no benefits to defence is:

> ***Marginal* expected damage assuming maximal AI elite exploit uplift to threat actor$_i$**
> = ((Capability$_i$ | AI uplifts threat actor$_i$ to find elite exploits) * Willingness$_i$ * Damages) - Baseline damages$_i$

Specifically, the AI uplift scenario we consider is defined by the following model evaluation result:

> ***AI elite exploit uplift:*** A study conducted at the end of 2025 finds that access to frontier AI models enables 25% of TA2 actors to find vulnerabilities and write elite exploits, assuming three months of full-time effort.

By default, we assume that frontier models are open weight and have no deployment safeguards. Call this Policy 0.

> **P0: Open weight with no refusals**. The model with elite exploit capabilities is open weight, and is released without deployment safeguards or other risk mitigations.[75]

[75] We describe this release policy in more detail in Appendix A.12.

For the reasons outlined in Section 5, we think it is plausible that, conditional on AI uplift, the risk increases in the short-term (over 2-3 months), but declines in the long-term (after 3-12 months) though it is unclear when exactly the risk would start to decline. To account for benefits to defence, we first calculate the increased risk conditions on AI uplift over a year assuming no benefits to defenders. We assume that AI uplift would start to reduce risk after around 6 months, and we roughly proxy this simply by dividing the expected costs over a year by two.[76]

Table 6.1 collates the parameter estimates from the other sections to produce baseline damages and damages conditional on this model evaluation result.

[76] For AI developers managing misuse risks, there may be other reasons, independent of offence-defence effects, to focus on damages only up to 6-12 months after release (see Appendix A.12).

Table 6.1. Upper bound expected costs of different levels of AI uplift over a year (assuming no benefits to defence)

Threat Actor Type:	**Inputs**				**Calculated social damages**	
	Capability		**Willingness:** Annual probability at least one actor in each class launches an attack	**Severity:** Damages from a data damaging worm attack	**Annual Baseline Damages:** {Severity} * {Capability} * {Willingness} Best guess [95% range]	**AI elite exploit uplift scenario:** counterfactual expected damages from AI uplift, accounting for benefits to defenders. Best guess [95% range]
	Exploit capability: Probability a random actor in each class can develop elite exploits	**Non-exploit worm capability:** probability a random actor in each class can write a worm if they already have elite exploits				
TA1: Individual hobbyist hacker	Very remote chance (~0%) *Significant uplift needed*	Extremely unlikely (0.1%-2%) *Significant uplift needed*	Extremely likely (~100%) *Many actors, one will do*	$01B–$100B	~$0	+ $18M [$1M-$200M]
TA2: Individual professional hacker	Very remote chance (~0%) *Significant uplift needed*	Unlikely (1–20%) *Significant uplift needed*	Highly likely (70-100%) *Many actors, one will do*		$375K [$10K-$10M]	+ $2B [$350M-$15B]
TA3: Team of 10 experienced hackers	Very unlikely (1-10%) *Significant uplift needed*	Realistic possibility (5-60%) *Some uplift needed*	Highly unlikely (1–5%) *Small subset may want to*		$5M [$100K-$200M]	+ $430M [$85M-$2B]
TA4: Team of 100 state-level hackers	Realistic possibility (5-60%) *Some uplift needed*	Almost certain (~100%) *No uplift needed*	Highly unlikely (1–5%) *Small subset may want to*		$170M [$10M-$3B]	+ $300M [$90M-$1B]
TA5: Team of 1,000 state-level hackers	Almost certain (90%–100%) *No uplift needed*	Almost certain (~100%) *No uplift needed*	Very remote chance (0.5%-1%) *No strategic incentive*		$210M [$45M-$1B]	+ $3M [$2M-$5M]
Total					$390M [$55M-$4B]	+ $2.7B [$530M-$18B]

The calculations for these estimates are in the ‘Expected damages AI uplift tab of’ Worm damages.

It is important to note that these estimates do not necessarily reflect the expected costs of *likely improvements* in AI exploit development capabilities. As a heuristic: it costs on the order of $1M-10M to develop an elite exploit, but TA2 actors have a budget of $10K for a specific attack. So, if AI did uplift TA2 actors, this would be comparable to an increase in their effective budget of 2-3 orders of magnitude. This sets a high capability bar and it may be unlikely that it is crossed in the near future. Our aim is to help to define a capability threshold, not to forecast how likely that threshold is to be crossed.

Table 6.1 shows that the expected damages of AI uplift to different threat actors are highest for TA2 actors and decline as we move up through actor skill levels. Uplift to TA2 actors would pose the greatest risks. Qualitatively, this is because (1) clearly some TA2 actors would be willing to launch data damaging worms if they could, and (2) AI uplift would make a large counterfactual difference to their ability to launch such attacks, as they clearly lack the capability at the moment. TA1 actors would also be willing to launch attacks, but are unlikely to succeed even conditional on the hypothetical AI uplift scenario.

At the other end of the scale, the risks of uplift for TA5 actors are small because (1) these actors are very likely already able to launch data damaging worms, and (2) they seem very unlikely to actually launch one.

All of these damage estimates are upper bounds because they assume that AI does not benefit defenders.

6.2. The effects of different risk mitigation policies

The scenarios above assume that models are released open weight without any deployment safeguards. Two alternative policies are:

> **P1: Proprietary models with refusals and anti-jailbreak measures.** Frontier AI models are proprietary and require users to access them via APIs with deployment and security safeguards. Companies train the models to refuse to respond to requests to find vulnerabilities or develop exploits, and have protections against jailbreaks. The models are assumed to be protected by SL-2 level infosecurity, which can likely thwart many moderate-effort attacks by individual hackers (RAND (2024)).
> **P2: Temporary protections with early access for defenders.** The public release of the model has P1 level safeguards in place.[77] However, a specific set of ‘cyber defenders’ is given access to a version of the model without refusals, i.e. with full vulnerability discovery and exploit development capabilities. The cyber defenders include four major software companies - Microsoft, Meta, Apple, and Google - and three highly

[77] For discussion of a similar idea, see Ee et al ‘Asymmetry by Design: Boosting Cyber Defenders with Differential Access to AI’ (2025).

vetted bug bounty organisations - Synack Red Team, Cobalt Core, and HackerOne Clear. After four months, the model is released under P0 security, i.e., is open weight.[78]

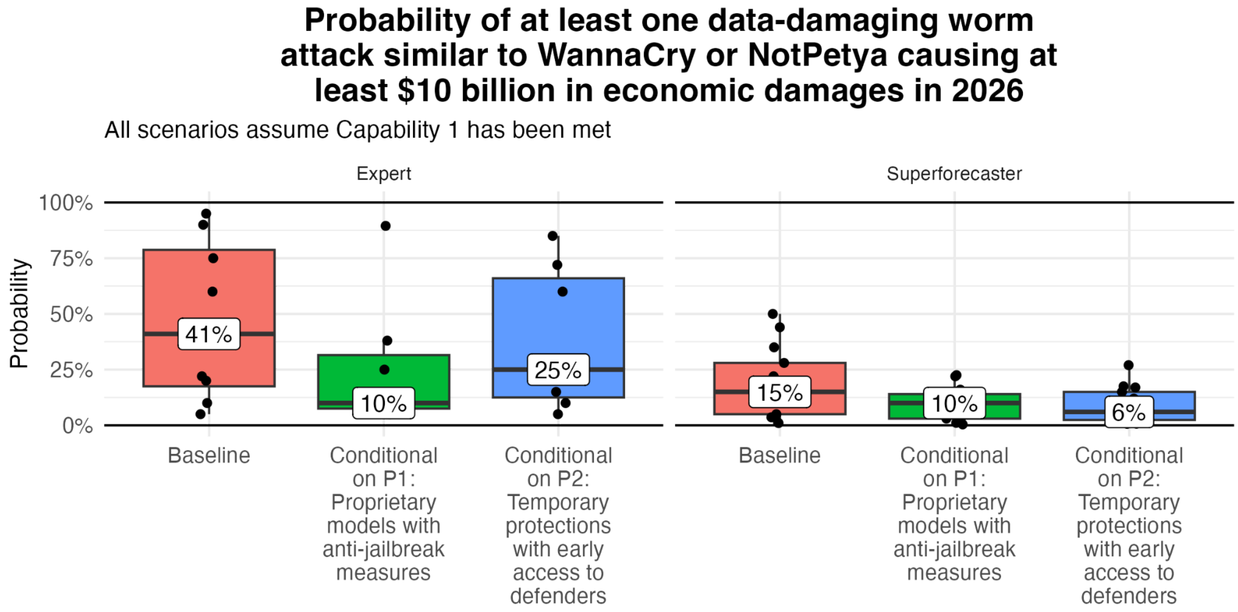
P2 is similar to responsible disclosure of vulnerabilities (where vulnerabilities are disclosed to the vendor and then declared publicly after a few months), but in this case applied to a vulnerability discovery *tool*, rather than to specific vulnerabilities.

We define these different policies in more detail in Appendix A.11.

These three policies each have different advantages and disadvantages. P0 – open weight models with no refusals – allows both attackers and defenders to benefit in full from AI vulnerability discovery and exploit development capabilities. P1 – proprietary models with refusals and anti-jailbreak measures – makes it harder for *both* attackers and defenders to benefit from these capabilities. Finally, P2 – temporary protections with early access for defenders – aims, over a short period, to limit benefits to attackers and provide benefits to defenders.

Which policy is optimal depends in part on offence-defence balance in this domain. Since, or so we have argued, it is unclear how AI uplift in vulnerability discovery and exploit development would affect offence-defence balance over time, it is also unclear which of these policies would be optimal. In our small pilot survey, we surveyed experts on the effect these three policies would have on the risk of data damaging worms. The results are shown in Figure 6.1.

Figure 6.1. Survey results on the effect of different risk management policies on the risk of data damaging worms, conditional on AI elite exploit uplift.



[78] P0 and P2 could be combined with subsidies for vulnerability discovery. More research is needed on what level of funding would be optimal, but for context, Apple, Google and Microsoft collectively pay on the order of tens of millions of dollars each for bug bounties (Google 2025; Apple 2022, Microsoft 2024).

As Figure 6.1 shows, there was no clear consensus on the effectiveness of the different policies, and there was a disagreement within and between experts and superforecasters.

One additional benefit of P0 and P2 compared to P1 – proprietary models with refusals – is that they might help to gather better information about model capabilities. There is evidence that model evaluations under-elicit true model capabilities, as they fail to capture how creatively humans might use models in the real world, by variations in prompting and scaffolding (Google Project Zero 2024).

Giving defenders early access to models or open sourcing models, and incentivising vulnerability discovery, might provide better information about model capabilities than traditional task-based evaluations.

Overall, more research on the merits of these different approaches may be warranted.

7. Conclusion

The rapid advancement of AI capabilities has prompted widespread concern about potential dual-use applications in cybersecurity, yet the policy discourse has lacked rigorous quantitative threat models to ground these discussions. This report addresses a critical gap by providing the first in-depth published AI-cyber threat model, offering a systematic framework for translating AI capability evaluations into concrete risk assessments for cybersecurity threats.

Our analysis reveals that if AI systems enable TA2 actors – individual professional hackers – to discover vulnerabilities and develop elite exploits, the expected annual damages could range from hundreds of millions to approximately ten billion dollars.

The central scenario examines a world where AI systems provide individual hackers with capabilities currently reserved for the most elite cybersecurity professionals and nation-state actors. In this scenario, our modeling suggests that the increased frequency and sophistication of attacks could impose substantial costs on the global economy. However, these estimates come with significant uncertainty. The modeling relies on analyses of threat actor behavior, technical capabilities, and economic damages. Judgments about these domains must to some extent be subjective and uncertain, as past empirical evidence is limited, and much relevant information may be classified.

Whether the estimated risk levels are unacceptable, and ought to be mitigated by AI companies prior to deployment, depends on societal risk thresholds. OpenAI uses a societal risk threshold of $100B in its Preparedness Framework, but there may be reasonable disagreement about this. Risk mitigations should take into account not only the magnitude of estimated damages, but the benefits and risks of AI's positive applications in cybersecurity defence and other domains.

More research should consider the comparative merits of different approaches to risk mitigation – open weighting, deployment safeguards, or early access to defenders. Due to the offence-defence balance in cyberspace, deployment safeguards like refusals have greater trade-offs than in other domains, but it remains unclear which policy would most reduce risk. The temporal dimension of cyber risk also matters for policy design, as AI-enhanced offensive capabilities may create particular vulnerabilities during the window between vulnerability disclosure and widespread patch deployment.

This report demonstrates the feasibility of applying quantitative risk assessment methodologies to AI dual-use concerns, providing a template that could be extended to other threat domains. The approach integrates multiple lines of evidence – technical analysis, historical case studies, and expert surveys – within a structured framework that makes assumptions explicit and enables systematic sensitivity analysis.

However, we view these results as a starting point rather than definitive estimates. Future work should prioritize larger-scale expert surveys to develop more robust consensus estimates and reduce the influence of individual forecaster biases.

The quantitative threat modeling approach developed here could be applied to other AI misuse risks. Each domain would require domain-specific technical analysis and expert consultation, but the basic framework – decomposing risk into more tractable components and surveying experts – appears broadly applicable. Such extensions would provide policymakers with more rigorous foundations for AI governance decisions across various threat domains and help prioritise risk mitigations.

The emergence of increasingly capable AI systems demands more sophisticated approaches to dual-use governance. This report demonstrates that quantitative analysis can inform these discussions despite the substantial uncertainties that remain. As AI capabilities continue advancing, threat modelling will be essential for ensuring that AI Safety Frameworks are grounded in the best available evidence.

Appendices

A.1. Wannacry and Notpetya timeline

EternalBlue and **EternalRomance** are exploits of vulnerabilities in the Server Message Block protocol used by many Windows systems prior up to 2017. **Doublepulsar** is a backdoor implant tool, which can be used to load malware on to infected systems.

- **2012**: EternalBlue was first developed by the NSA by 2012 at the latest. The NSA did not disclose the vulnerabilities it exploited to Microsoft, and instead used the exploit in its own cyber operations.
- **December 2016 -** Sandworm starts work on the NotPetya worm (Maschmeyer 2021).
- **2016 - Early 2017**: At some point between likely between 2016 and early 2017, a hacker group known as the Shadow Brokers stole various hacking tools from the NSA, including EternalBlue, EternalRomance, and DoublePulsar.
- **Early 2017**: The NSA informs Microsoft of the vulnerabilities possessed by the Shadow Brokers (Ars Technica 2017)
- **February 2017:** Russian hackers responsible for the NotPetya attack may have had access to the Shadow Brokers exploits that were eventually leaked to the public in April (Ars Technica 2017)
- **14th March 2017**: Microsoft released a patch for these vulnerabilities (Microsoft 2017). This included a patch for Windows XP, which Microsoft had stopped supporting in 2014 (WIRED 2017).
- **14th April 2017**: The Shadow Brokers leaked the NSA hacking tools, including EternalRomance, EternalBlue and Doublepulsar (Ars Technica 2017).
- **11th May 2017**: Security researchers developed a scanner and exploit module for EternalBlue for Metasploit. (Github).
- **12th May 2017**: The WannaCry attack, which used the NSA tools EternalBlue and DoublePulsar, was launched, and infected more than 230,000 computers across 150 companies (Symantec 2018). Due to the kill switch, after 6 hours, no new systems were encrypted (Wired 2017).
- **14th May 2017:** Microsoft criticises the NSA for not promptly disclosing the vulnerabilities eventually stolen by the Shadow Brokers (Microsoft 2017).
- **27th June 2017**: The Notpetya attack, which used the NSA tools EternalBlue and EternalRomance, was launched. 75% of affected systems were in Ukraine (ESET 2017), but the attack caused collateral damage in numerous other countries, including Russia (Greenberg, *Sandworm*, p.198).

A.2. Shadow Broker prices

The Shadow Brokers tried to sell various stolen NSA hacking tools from 2016 through to early 2017. However, it was not known that they had EternalBlue, EternalRomance or DoublePulsar until April 2017, when those tools were leaked publicly.

- The Shadow Brokers publicly announced that they had various NSA hacking tools in 2016, and several experts suggested that the tools were legitimate NSA tools (Tripwire 2017). The Shadow Brokers tried to sell all of the tools in 2016, with a stated desired price of $560M (Tripwire 2017), but they only received bids of around $1K (Sophos 2016).
- The Shadow Brokers later tried to sell individual NSA hacking tools for hundreds of thousands of dollars in January 2017, but after receiving little interest, released a cache of Windows hacking tools (not including EternalBlue, EternalRomance or DoublePulsar) for free a week later (Tripwire 2017).
- The Shadow Brokers only announced that they had the tools used in the WannaCry and NotPetya attacks - EternalBlue, EternalRomance and DoublePulsar - in April 2017, but never tried to sell these tools, and instead gave them away for free (Ars Technica 2017; Ars Technica 2017).

There are various reasons that the auction price may not have reflected the true market value of the EternalBlue, EternalRomance and DoublePulsar. First, as noted, the Shadow Brokers never disclosed that they had those specific tools until they gave them away for free in April 2017. Buyers would therefore have needed to have paid for the tools without knowing what product they were receiving. Secondly, the auction required bidders to send bitcoin to the Shadow Brokers' address with no hope of getting the bitcoin back if they did not win the auction (Wired 2016). Third, it would also have been difficult for buyers to be sure that they would actually receive the tools if they did pay. Finally, given the public attention on the sale, buyers would also have faced significant attention from authorities for buying the exploits (Sophos 2016), which would have further reduced demand.

A.3. Technical details of how EternalBlue enabled the WannaCry and NotPetya worms

EternalBlue was used extensively in surveillance and counterterrorism operations by the NSA for at least five years (NYT 2019). However, the exploit could also be used in data damaging computer worms.

How EternalBlue worked

The SMBv1 protocol exploited by EternalBlue was first developed in 1983 (Avast 2020), and is now widely recognised as insecure. On many systems vulnerable to EternalBlue, port 445 (over which SMBv1 communicates) was exposed to the open internet, so a network scanning feature could quickly propagate the exploit to other systems. On newer systems, port 445 is not exposed to the internet. Microsoft deprecated the SMBv1 protocol in 2014 (Microsoft 2023), patches for the vulnerabilities exploited by EternalBlue have been available since 2017, and since 2017, SMBv1 has been disabled by default in Windows systems updates (Microsoft 2023). EternalBlue worked as follows:

1. **Establish a connection with Port 445.**
    a. The attacker sends a specially crafted packet to the target system's port 445, which is used for SMB communications.
2. **Malicious SMB Packets:**
    a. EternalBlue constructs malformed SMB packets that contain more data than the SMBv1 buffer can handle (Nguyen et al 2024)..
    b. These packets are designed to overflow a buffer in the SMB service, specifically within the function that handles the SMB transaction requests.
3. **Triggering the Buffer Overflow:**
    a. When the target system receives these specially crafted packets, it overflows the allocated buffer in kernel memory (Nguyen et al 2024).
    b. The buffer overflow allows EternalBlue to overwrite critical memory regions, including function pointers or return addresses.
4. **Executing Arbitrary Code:**
    a. A system module in Windows, the Hardware Abstraction Layer, uses a heap with a fixed address (Nguyen et al 2024)..
    b. EternalBlue uses a technique known as ‘heap spraying’ to place malicious payloads at predictable locations in memory.
    c. Once the buffer overflow is triggered, the attacker can write binary code that is executable on the heap of the Hardware Abstraction Layer (Nguyen et al 2024)
5. **System privileges:**
    a. Since the SMB service runs with system privileges (the highest level of privileges on a Windows system) the injected shellcode also executes with these high privileges, giving attackers full control over the compromised system, without the need for authentication.

A.4. Elite exploit prices

There are various different markets for exploits. The data we have on exploit prices in those markets suggest that the market price of elite exploits is on the order of $10M. Elite exploits are often sold to state intelligence agencies, suggesting that they are difficult to develop in-house, even for states.

Grey market broker prices

Grey market brokers like Zerodium and Crowdfense buy the code for exploits from individuals, groups or companies and then sell them to governments. Elite exploits sell on these platforms for $1m to $9m. Broker markets are inefficient due to information asymmetries between buyers and sellers, and is therefore subject to adverse selection, or the “market for lemons”.

> “The zero-day exploit market is a market with extreme information asymmetries. The seller has much more information about whether the exploit is actually working. The market is also flooded with lemons. Many of the exploits offered are a lot less reliable than sellers initially report. Also, the buyer of an exploit is not always able to test the exploit before purchasing it, as the economic value would be lost once given to the buyer for “testing.” This structural setup makes even beneficial zero-day transactions difficult.” (Smeets 2022).

If buyers are unable to tell the difference between strong and weak exploits, they would be unwilling to pay high prices. Consequently, the price is bound to be lower than what sellers of high-quality exploits would sell for, driving them out of the market.

Prices on exploit broker platforms have increased in recent years: in 2019, the highest bounty offered by Crowdfense was $3M (Tech Crunch 2024), whereas today the highest is $9M. Many cyber experts attribute this to companies hardening their products against hackers (Tech Crunch 2024).

Commercial surveillance vendors

Commercial surveillance vendors like NSO and Intellexa typically charge hundreds of thousands of dollars *per device* infected by elite exploits and the spyware payload installed by those exploits.[79] For a single customer, the basic package usually includes 10-20 devices each using elite exploits. This suggests that the value of the elite exploits and the spyware for an individual customer is on the order of millions of dollars. The commercial surveillance vendors would also have numerous customers for the same product, which suggests that the value of individual elite exploits plus the spyware is at least tens of millions of dollars.

It is difficult to know what fraction of the value of the product comes from the elite exploits that allow full control of the device, and what fraction from the spyware payload. Still, it seems plausible that the value of the exploits alone is well in excess of $1M, and likely on the order of $10M.

Moreover, these tools are sold to some state intelligence agencies (Google Threat Analysis Group, *Buying Spying*, 2024), which suggests that they are difficult to develop even for state intelligence agencies.

[79] Commercial Surveillance Vendor product prices are collected together in:
Exploit prices from commercial surveillance vendors

Bug bounties

Apple offers $100k to $1M for zero-click remote access exploit chain with full kernel execution and persistence (i.e. the exploit continues to work after the device has been rebooted) on a single recently released device. Google offers $1m for a zero-click RCE with persistence exploit that is effective against all vulnerable builds and models of Pixel Titan M. The price for vulnerabilities affecting a large number of distinct systems would be much higher. As noted in the main report, WannaCry was effective against 400M systems at the time of the attack. This suggests that in today's prices, EternalBlue would be worth millions of dollars, and plausibly on the order of $10M.

In 2016, Google's Project Zero ran a 6 month public competition for hackers to find a full exploit chain that achieves remote code execution on multiple Android devices knowing only the devices' phone number and email address, and provides access to third-party application files in internal storage (Contest rules). This would qualify as an elite exploit. The first prize of $200k, second prize of $100k, and a third prize of $50k (Prize announcement). There were no viable entries for the competition (Post-mortem). This is directional evidence that finding such exploits is difficult, and suggests that the cost of finding such exploits is *at least* tens of thousands of dollars.[80]

A.5. FORCEDENTRY elite exploit and possible worm application

Over the last few years, many of the elite zero-day exploits discovered being used in cyberattacks were developed by NSO Group, an Israeli commercial surveillance vendor.[81] To illustrate the complexity of newer elite exploits, I will briefly discuss NSO's FORCEDENTRY exploit chain. FORCEDENTRY could be used to install Pegasus spyware on to an iPhone, which can access messages and files, steal emails, steal contact information, tap the microphone and camera, track location, exfiltrate data, alter settings, and turn functions on and off, amongst other things.

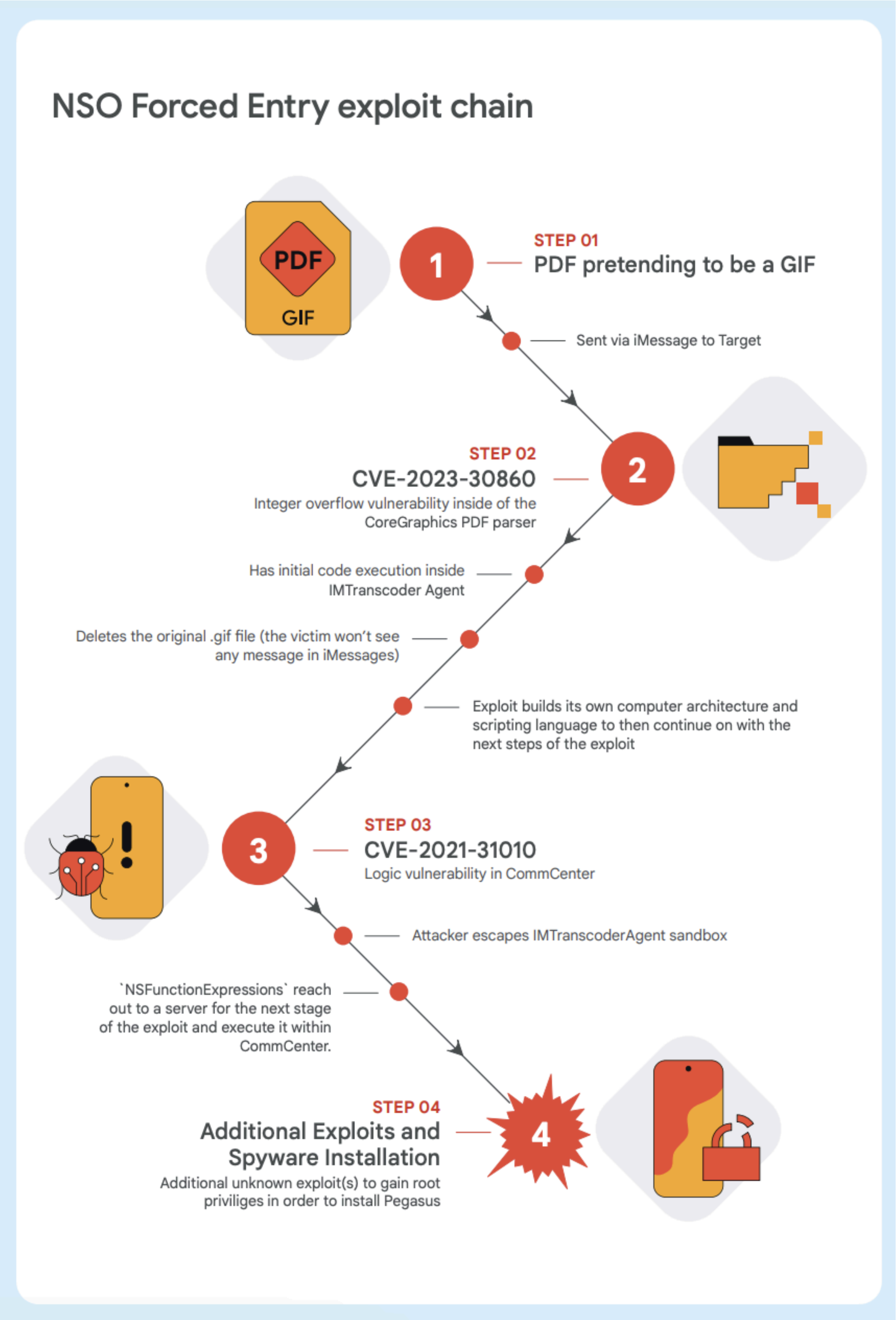
The FORCEDENTRY exploit could install Pegasus on iPhones without any interaction from the user (Google Threat Analysis Group, *Buying Spying*, 2024, pp. 36-37; CitizenLab 2023). It gains initial access with a remote code execution vulnerability in iMessage, a user space application. Every iOS app runs in its own sandbox. Even if an attacker manages to compromise an app like iMessage, the malicious code is confined to the sandbox and cannot directly affect other apps or the core system. Consequently, FORCEDENTRY also included a sandbox escape exploit, as well as a local privilege escalation exploit that was required to give full root access to the device,

[80] The value of the prize money should be discounted by: the *ex ante* chance of winning the competition; risk aversion; and the opportunity cost of entering the competition.

[81] Many other commercial surveillance vendors develop 1-click exploits, which require some interaction on the part of the user (Google Threat Analysis Group, *Buying Spying*, 2024). 1-click exploits are less suited to fast spreading worms for this reason.

allowing the malware to install the Pegasus spyware (Google Threat Analysis Group, *Buying Spying*, 2023, p. 36). The figure below summarises the FORCEDENTRY attack chain.

Figure A.1. NSO FORCEDENTRY exploit chain



NSO Forced Entry exploit chain

STEP 01

PDF pretending to be a GIF

Sent via iMessage to Target

STEP 02

CVE-2023-30860

Integer overflow vulnerability inside of the CoreGraphics PDF parser

Has initial code execution inside IMTranscoder Agent

Deletes the original .gif file (the victim won't see any message in iMessages)

Exploit builds its own computer architecture and scripting language to then continue on with the next steps of the exploit

STEP 03

CVE-2021-31010

Logic vulnerability in CommCenter

Attacker escapes IMTranscoderAgent sandbox

`NSFunctionExpressions` reach out to a server for the next stage of the exploit and execute it within CommCenter.

STEP 04

Additional Exploits and Spyware Installation

Additional unknown exploit(s) to gain root priviliges in order to install Pegasus

In contrast, EternalBlue directly exploits an SMB vulnerability at the kernel level, which operates outside of the typical user-space sandbox. The table below summarises the difference between Eternal Blue and the zero-click user-space exploit used for initial access in FORCEDENTRY.

Table A.1. Differences between EternalBlue and user-space iMessage exploits

Feature	**EternalBlue (SMB RCE)**	**Zero-Click User-Space Exploits (e.g., iMessage, browsers)**
Target Level	Kernel-level	User-space applications
Sandbox Involved?	No	Yes
Privileges Gained	SYSTEM	Initially limited, requires escalation
Interaction Required	None (remote exploit)	None (zero-click), but needs sandbox escape

A data damaging worm using FORCEDENTRY could proceed as follows:

Step 1: Initial Infection via FORCEDENTRY

- An attacker uses a FORCEDENTRY to remotely compromise a device via iMessage.
- The payload could install a malicious program that runs with elevated privileges and gains complete control of the device, gaining access to the device's contacts, messaging apps, and network information.

Step 2: Self-Replication Mechanism

- Once the device is compromised, the worm could scan for other vulnerable targets.
- It could leverage FORCEDENTRY to send malicious exploits to contacts using iMessage, or compromise devices on the same network.
- The worm would continue to replicate itself on each new infected device, spreading exponentially.

Step 3: Data encryption

- The worm could wipe or encrypt data on the device.

However, it is plausible that a worm propagating in this way would be noticed by modern cybersecurity systems as devices sending out messages in bulk would be flagged as suspicious activity and stopped.

A.6. Elite exploits discovered being used in cyberattacks since since 2017

Google's Threat Analysis Group maintains a Google sheet of zero-day exploits publicly discovered being used in cyberattacks since 2014. The sheet does not specify whether any of the exploits were elite-level. I performed my own analysis on which of the exploits in this dataset were plausibly elite-level.

The first method involved inputting all of the vulnerabilities in the sheet into Claude 3.5 Sonnet and asking it to identify which of the exploits were elite-level, per my definition. The reliability of this method is open to question because Claude hallucinated certain details of the vulnerabilities mentioned. I manually checked the properties of the remaining exploits discovered since 2020. On this procedure, of exploit chains discovered being used in cyberattacks 2020-2023, only the following are elite-level:[82]

- Various exploit chains of iOS used for the installation of NSO's Pegasus spyware.
  - These exploit chains include:
    - BLASTPASS, 2023 (CitizenLab 2023)
    - FORCEDENTRY, 2021 (Google Threat Analysis Group, *Buying Spying*, 2023, p. 36)
  - Each of these exploit chains gained an initial access by exploiting zero-click vulnerabilities in iMessage.
- Triangulation malware
  - Likely developed by the US and/or its allies, also involves a 0-click exploit that gained initial access by exploiting an iMessage vulnerability (Kaspersky, Operation Triangulation).
- The vulnerabilities used in the 2021 Microsoft Exchange attacks by Chinese APTs.
  - The vulnerabilities and exploits for this attack were found by a team led by Orange Tsai, a 3-4 person team who appear to be among the best white hat hackers in the world.[83]

For the second method, I focused on exploits only of Microsoft products from 2020-24. The rationale for this is that Microsoft products are widely used and have been involved in past serious cybersecurity incidents, and so they seem to pose the greatest risk of data damaging worm attacks. I fed the Microsoft zero-day vulnerabilities in the Google Threat Analysis Group database into ChatGPT 4o and asked whether any of them allowed remote code execution with kernel level privileges. I then manually checked whether any of the remaining vulnerabilities were elite-level in the sense that they were used in exploit chains that did not require user interaction. This method suggests that the vulnerabilities exploited in the 2021 Microsoft Exchange attacks were the only elite exploits.[84] Indeed, Microsoft itself classified some of the

[82] The results are in the 'all elite level exploits' filter view in the 'Claude 2021-24 CVSS and exploit chain' tab of this sheet.

[83] Orange Tsai's team includes three to four people (Zero Day Initiative 2021). They have has won awards including "Master of Pwn" at Pwn2Own 2021 and 2022. Their research earned him the Pwnie Awards winner for "Best Server-Side Bug" in 2019 and 2021 and also secured 1st place in the "Top 10 Web Hacking Techniques" for 2017 and 2018 (Orange Tsai, nd; Hacktivity nd). One expert told us that Orange Tsai is one of the best hackers in the world.

[84] See the 'Microsoft elite-level exploits' filter view in the third tab of this sheet.

exploits for the Exchange attacks as wormable, as no human interaction would be required for an attack to spread from one vulnerable Windows box to another (Krebs 2022; Microsoft 2022).

This method may fail to identify some elite exploits. In some cases, the full exploit chain used in an attack is not recovered. So, even though individual exploits were not known to be used in elite exploit chains, and did not by themselves have elite capabilities, they may nevertheless have been used in elite exploit chains. For Microsoft vulnerabilities for instance, many of the known zero-days being used in cyberattacks were local privilege escalation exploits, which could potentially allow remote code execution with kernel level privileges if combined with an RCE exploit for initial access.

A.7. Motivations and functionality of past worm attacks

This section discusses details of the past worm attacks in the Johansmeyer dataset. Table A.2 outlines the number of devices infected by each worm and the source for this estimate.

Table A.2. *Number of systems infected by past significant worms*

Attack name	# systems Infected	Source
Melissa	~100K	GAO 1999
ILOVEYOU	~50M	Dark Reading 2020
Klez	~7M	HP 2020
CodeRed	~360K	CAIDA 2002
Nimda	>1.3M	CNET 2001
SirCam	~2.3M	Wired 2003
SoBig	>1M	Al Jazeera 2003
SQL Slammer	>75K	CAIDA 2003
Swen	~1.5M	SmarterMSP 2021
Mimail	~21K	SiliconRepublic 2003
Yaha	?	No infection # found
MyDoom	~500K	NBC 2004
Sasser	~500K-1M	NBC 2004; Guardian 2004
StormWorm	1M-50M	Wired 2007
Conficker	~10M	Guardian 2009

WannaCry	~230K	Symantec 2018
NotPetya	~670K	See fn[85]

Worms with an unknown creator

There were two versions of the **CodeRed** worm which each exploited a buffer overflow vulnerability in Windows servers to deface sites using the server with ‘Hacked by Chinese!’, and launch denial of service attacks against specific websites, including the website of the US President (Kaspersky 2022), though it did not destroy data on infected systems (GIAC 2005). A patch had been available for the worm a month prior to launch (Microsoft 2001; GIAC 2005). Code Red was a network worm that spread without user interaction, allowed remote code execution with system-level privileges (GAO 2001; GIAC 2005). However, due to its limited reach, the CodeRed exploit does not count as ‘elite’ per our definition. There is evidence that the worm was launched from a Chinese university, though the perpetrator is not conclusively known (CAIDA 2002).

The **Nimda** worm targeted Windows systems and spread via email, web servers, web browsers and shared network drives (GIAC 2002). Nimda could spread without user interaction, allowed remote code execution (Govexec 2001; GIAC 2002), but given its limited reach, the exploit used does not count as elite. Nimda primarily caused damage via slowing down networks, but it also gave attackers elevated privileges on infected systems (GIAC 2002). It is unclear what the motivations for the attack were (GIAC 2002).

The **Swen** worm targeted Windows systems and replicated via email, local network, IRC and file sharing websites, and targets Windows systems (Microsoft 2005). It used a vulnerability in Internet Explorer to execute directly from email (F-Secure, nd). The worm required the user to open an attachment, and so was no zero-click and therefore did not involve an elite exploit (Microsoft 2005; F-Secure, nd). The worm aimed to terminate the processes of antivirus software and firewalls, making systems vulnerable to other malware, but did not attempt to deploy any further destructive payloads (F-Secure, nd). It could also potentially cause damage by creating high network traffic (Comodo 2019).

The **Mimail** worm spread via infected email attachments (F-Secure), and therefore did not use an elite exploit. It appears to have been launched for financial reasons. It was designed to infect a large number of systems to launch denial of service attacks against anti-spam organisations, and may have been launched or funded by spam groups (Wired 2003), while some variants attempted to steal credit card information (F-Secure).

[85] For NotPetya, according to one source, there were 500K infections in Ukraine alone (Maschmeyer 2021). 75% of infections were in Ukraine (ESET 2017), which implies ~670K infections globally.

Worms launched by OC1 and OC2 actors

The **Melissa** worm caused damage by creating large amounts of network traffic, and did not contain a malicious payload. The worm required users to click a link in order to spread (FBI nd) and therefore did not involve a zero-click elite exploit. The perpetrator claimed that the worm was not designed to do as much damage as possible, and that the damage was accidental (Register 2001; ZDnet 1999), though it is unclear if this is true.

iLOVEYOU was a worm that spread primarily via email and required users to open an attachment (Kaspersky 2022), and therefore did not use an elite exploit. The worm irretrievably corrupted documents on infected systems, and sent passwords to the creator. Later variants completely wiped the hard drive (Kaspersky 2022). The attacker apparently released it to prove his own hacking skill (NYT 2000).

The **SirCam** worm targeted Windows and spread via email and required users to download an attachment with malicious code (CNET 2001), and therefore did not use an elite exploit. Because it attached actual user files to emails, the worm risked exposing confidential information. Its payload also attempted to delete all files on a system in certain conditions (CNET 2001) and fill up the drive where Windows is installed , though this did not work in practice due to a bug in the code (F-Secure). It is unclear what the motivations for the attack were (CNET 2001).

Sasser caused damage by creating a large amount of network traffic, which caused many companies to shut down their systems (NBC 2004; Tech Monitor 2014). The exploit used by Sasser should plausibly be classed as ‘elite. The worm could spread by exploiting the operating system through a vulnerable network port, and so could spread without user intervention, but it could be stopped by a properly configured firewall or by downloading system updates from Windows Update (Tech Monitor 2014). The vulnerability exploited also allowed remote code execution with system level privileges (Microsoft 2004). Sasser infected 500K to 1M systems, and was released after the patch for the vulnerability was released. If it has been released prior to the patch release, likely many more systems would have been vulnerable. Authorities suspected that the author released it to prove his hacking skill (NBC 2005).

Klez was a worm that caused damage by creating a large amount of email traffic, disabling antivirus and security software, and potentially by sharing confidential information in attachments to emails (Microsoft 2007). The virus required users to download an email attachment, and therefore did not use an elite exploit. It was only effective against systems that had failed to install a patch that had been available for a year (Wired 2002). The perpetrator and motivations are unknown.

SoBig was an email worm that used infected devices to distribute spam, apparently for financial gain (NBC 2003). The worm caused damage by creating large amounts of network traffic, causing company systems to slow down or stop (CNN 2003). The worm required users to open an attachment in an email (NBC 2003), and therefore did not use an elite exploit.

SQL Slammer did not contain a directly destructive payload, but caused damage by creating large amounts of network traffic (CAIDA 2003; ThreatPost 2010). A patch for the vulnerability exploited by SQL Slammer had been available for six months before the worm was released (Wired 2003). It is unclear what the motivations were for the attack. SQL Slammer did not require user interaction to spread, but the worm did not use an elite exploit because it was effective against far fewer than 10 million devices: it infected 90% of vulnerable hosts within ten minutes, and infected at least 75,000 devices (CAIDA 2003).

MyDoom was an email worm that tried to create a large botnet of infected computers to launch denial of service attacks against specific companies including SCO and Microsoft, while also slowing down infected machines (Okta 2024). Since the worm required users to download an infected attachment to spread (Okta 2024), it did not use an elite exploit . Some argue that the worm was launched by disgruntled members of the open source software community, but the worm may also have been launched for financial reasons (WIRED 2004; Al Jazeera 2004; New Scientist 2004; Network Computing 2004).

Worms launched by OC3 actors

Yaha was released by the Indian Snakes, a group of Indian hackers in order to retaliate against Pakistani hackers who had defaced Indian websites. The worm attempted to launch denial of service attacks against Pakistani websites (F-Secure nd; Help Net Security 2002; Wired 2003). In order to spread, it required users to click infected links (F-Secure nd), and so it did not use an elite exploits.

StormWorm was released by Russian Business Network, a Russian cybercrime group. Storm Worm charged a fee for denial of service attacks against anti-spam websites and security vendors (Hypr nd). Storm worm spread via email and social engineering (Hypr nd), so it did not use a zero-click elite exploit.

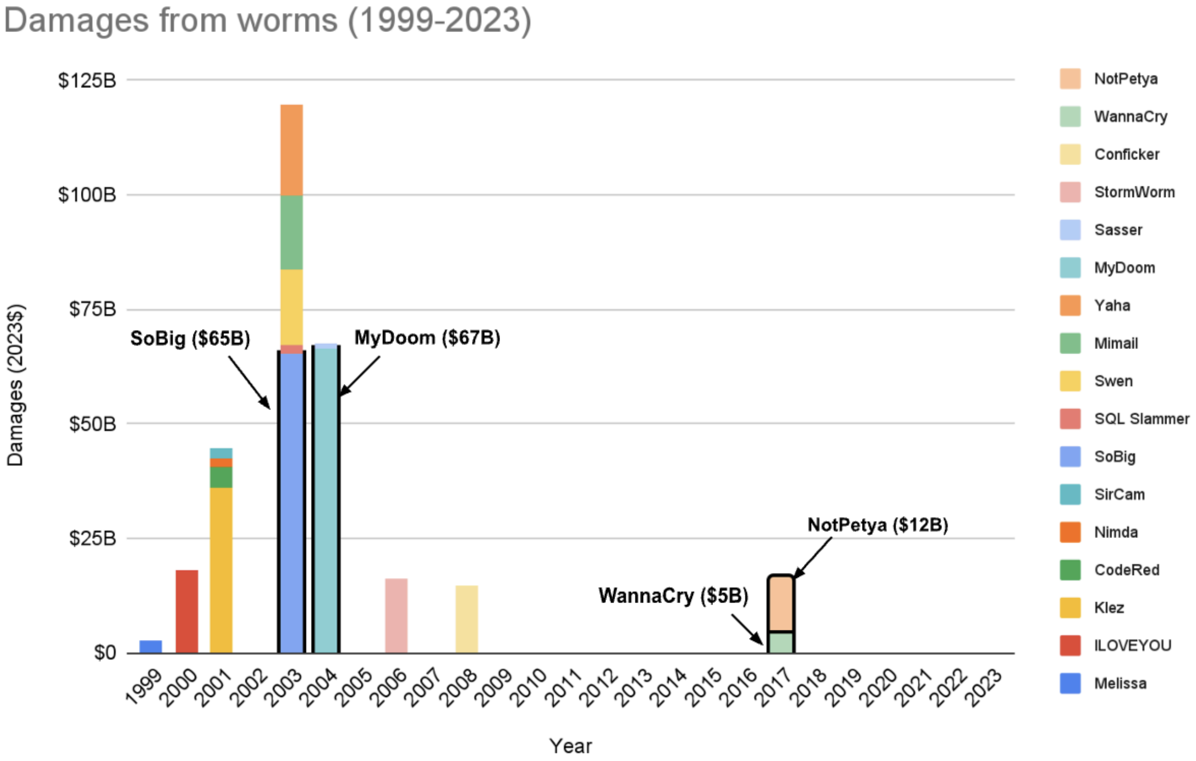
Conficker was produced by a Ukrainian cybercriminal group. It infected millions of computers, but was never used for significant malicious attacks due to concern about criminal repercussions (NYT 2019). Conficker was zero-click, spreading via network shares, allowed remote code execution (CAIDA 2009; CISA 2013) with system privileges (Microsoft 2008), and was plausibly effective against more than 10 million systems.

A.8. A critique of estimates of past cyber damage estimates

The figure below shows damages in each year from worm attacks only, according to the Johansmeyer (2024) data.[86]

[86] As noted in the main report, this excludes Stuxnet.

Figure A.2. Damages from major worm attacks in Johansmeyer dataset (1999-2023)



Johansmeyer notes that the provenance of this data is poor.

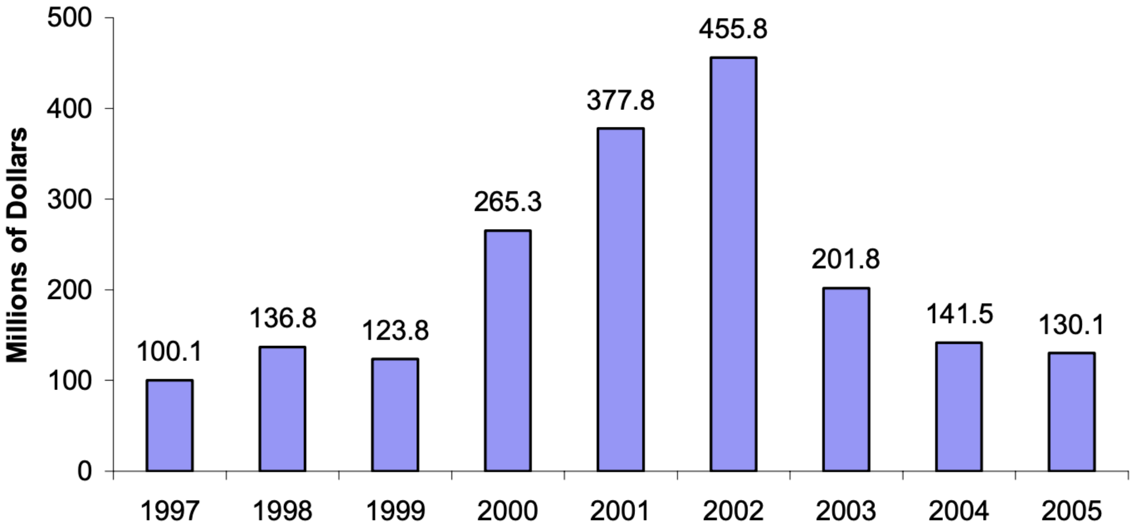
> “To arrive at 21 events since 1998, I pulled data on historical economic losses from cyber catastrophes from publicly available sources, which is the primary limitation of the study. Unfortunately, many estimates come from popular media sites and corporate blogs… Methodological information for the publicly available estimates is virtually non-existent, and some sites presumably reference (but don’t link to) long-gone sources. Essentially, I'm relying on judgment and comparison of original estimates without a benchmark to make the best of a bad situation. That said, many of the estimates appear to have been recycled and republished, which offers at least a veneer of respectability.” (Johansmeyer nd)

Moreover, Johansmeyer (2023), fn 1 notes that where there are multiple sources, he chooses the higher estimate. (This is because Johansmeyer’s aim is in part to show that even on the high estimates, cyber risk is lower than many people argue.) For instance, some of the estimates come from a November 2003 House Committee Hearing on computer viruses. In that hearing, different speakers and sources gave damage estimates for SoBig ranging from $500m to $30B. One source used in the dataset reports that the Melissa worm caused >$1B in damage (in 2012 dollars) (Beattie 2012), but in the plea agreement, the Melissa perpetrator admitted to causing “over $80M” in damage (The Register 2001).

The original source for many of the very large pre-2005 estimates is Mi2g, which have been heavily criticised for producing greatly inflated estimates (Leyden, 2002; RTI 2006). RTI 2006 argues that Mi2g had a financial incentive to produce overinflated estimates in order to attract media and customer attention. Mi2g used a proprietary model that cannot be publicly scrutinised. The Mi2g data implies that between 1999 and 2003, there was a 46X increase in the damages from significant cyber attacks.

In the same period, the CSI/FBI cybercrime survey found a 1.6X increase (RTI 2006 sec. 2.2.). The CSI/FBI Computer Crime and Security Survey represents the responses of hundreds of IT professionals in U.S. corporations, financial institutions, government agencies (federal, state, and local), medical institutions, and universities. Participants were surveyed to determine the spending of their organizations on cyber security, the number of breaches and the associated financial losses incurred during the previous year, and the preventative activities undertaken. The results of the survey for 1997 to 2005 are shown below:

Figure A.3. Costs of cybercrime according to the CSI/FBI cybercrime survey (1997-2005)



Source: RTI 2006 p. 19.

The CSI/FBI survey is widely referenced by academics, government agencies, and companies providing security-related products or services. However, the authors recognise that the survey data is not comprehensive, as noted by a report for the Airforce Research Organization

> “However, 20 percent of the responding organizations acknowledged that they do not report all computer intrusions to law enforcement because of the high cost of doing so. Furthermore, cost-estimating procedures are not uniform; capturing labor resources allocated to security or employee productivity loss is not easy and is not always consistent. Thus, the authors acknowledge that the information garnered by this survey, while accurate as reported by respondents, should not be considered a complete accounting of the costs of cyber security.” (RTI 2006 p. 19)

Thus, the CSI/FBI survey is not a good estimate of the absolute costs of cyber attacks. However, it seems more reliable than Mi2g with respect to the change in the trend of costs over time. If we assume that the 1999 damage estimate in Johansmeyer's data is correct, and use the 1.6X increase ratio implied by the CSI/FBI data, then the cost of significant cyber attacks in 2003 would be $4B, rather than the $120B implied by the Mi2g data.

Overall, we would not be surprised if the pre-2005 damage estimates were too high by one or more orders of magnitude.

Estimates of WannaCry and NotPetya in the Johansmeyer dataset

The damage estimates for WannaCry and NotPetya in the Johansmeyer dataset also seem unreliable. Different sources give different estimates for the costs of WannaCry, with some claiming that the costs were $4B (2017$) (Johansmeyer (2024)), and others claiming damages of $8B (2017$) (Insurance Business Mag (2017)). The source for both estimates is the cyber insurance company Cyence, but I have been unable to find the original source, or the calculations for this estimate.

The costs of NotPetya were estimated to be at least $10B (2017$) (WIRED 2018; Greenberg, *Sandworm*, p. 215).[87] The source of the NotPetya estimate is a claim by former Homeland Security adviser Tom Bossert, who at the time of the attack was President Trump's most senior cybersecurity-focused official (WIRED 2018). However, no public calculations are available for this estimate.

A.9. Crosignani et al estimate of NotPetya costs

This section discusses the Crosignani et al (2023)[88] estimate of damages from WannaCry and NotPetya in more detail.

Crosignani et al searched for firms affected by NotPetya by (1) web scraping SEC filings in 2017 and 2018; (2) searching newspaper articles; and (3) checking against reporting in Greenberg's *Sandworm* book. They exclude firms in Ukraine, Russia, and "non-public firms that [they] would be unable to find in other data sets", such as government agencies and hospitals (p 6-8). They identified 8 affected firms, including Merck, FedEX and Maersk, and collated their reported costs (Table 1).

[87] These figures are in today's dollars.

[88] An open access working paper version of Crosignani et al (2023) is available here. Page references in this section refer to the working paper version.

The costs reported by the firms include (1) lost or delayed revenue, and (2) remediation costs, such as analysis of IT systems, replacing and repairing equipment, and restoring services. The total reported costs across the 8 firms were $1.8B (Table 1).

Crosignani et al (2023) also estimate the costs to upstream and downstream companies in the supply chain for the eight companies. They identify 233 customers and 320 suppliers indirectly affected by the cyberattack, i.e. exposed through their supply chain connections to directly hit firms (p. 9). They use a difference-in-differences approach comparing the change in performance of firms indirectly affected by the shock through their supply chain with that of unaffected firms operating in the same industry, country, and size quartile in the same year (p. 11). They estimate the effects on the ratio of earnings before interest and taxes to total assets for firms affected by the cyberattack compared to the control group.

They found that NotPetya had a statistically significant effect on downstream customers of directly affected firms, but not on suppliers of affected firms (p. 16-19). They found that NotPetya led to a 1.3 percentage point drop in the ratio of earnings before interest and taxes to assets of downstream customers. They note that a conservative estimate of the supply chain effects on customers suggests a drop in profits of $7.3bn (p. 18-19). Combined with the estimate of the direct costs of NotPetya, this implies total costs of $9.1bn in 2017 dollars, or $11.3bn in 2023 dollars.

Reasons the Crosignani estimate may be biased high

Crosignani et al (2023) may overstate the costs of NotPetya for two reasons. Firstly, our aim is to estimate the social costs of the NotPetya attack, but not all of the costs reported in Crosignani et al reflect true social loss. In a competitive market, lost revenue for particular firms can be offset by rival firms *gaining* revenue, as consumers switch to those rival firms. Therefore, the loss to directly affected firms could be offset by gains to rival firms, and the costs to consumers could be offset by switching to competitors' products. Neither of these offsetting effects are measured in Crosignani et al. This biases the direct cost estimates high.

However, remediation costs *are* pure social loss.

This problem also applies to the Crosignani et al calculations of costs to customers in the supply chain. They measure the economic costs by comparing a treatment group of customers affected by NotPetya to a control group in the same industry, country and size quartile not affected by NotPetya. The difference between the performance of the treatment group and the control group may in part reflect a pure loss to the treatment group, but may also in part reflect *gains* in the control group.

Second, some of the direct costs reported in Crosignani et al refer to sales being delayed,[89] but this again is something of a ‘soft’ metric, as the losses to producers and consumers could be offset by increased sales in subsequent periods.

Although these factors may bias the Crosignani et al estimate high, we do not think they would completely offset the reported damages. It is difficult to know how large the offsetting effects might be. We assume that this might justify a 0.5X adjustment to the reported overall damages in Crosignani et al (2023).

Calculations are in the ‘NotPetya & WannaCry damages’ tab of Worm damages.

Reasons the Crosignani estimate may be biased low

Crosignani et al may also be biased low because they do not consider the costs to companies in Ukraine, even though Ukraine accounted for 75% of infections (Eset 2017), and, according to one Ukrainian official, 5% of all private, corporate and government computers in Ukraine could not be repaired following the attack.[90]

The Ukrainian Finance Minister Oleksandr Danylyuk claimed that NotPetya cost 0.5% of Ukrainian GDP (Hromadske 2017), which implies losses of $560M (in 2024 dollars). This is much smaller than the total costs estimated by Crosignani et al.

This illustrates a more fundamental problem with using money as a metric to measure social costs. Ukrainian GDP per capita in 2017 was 10-20x smaller than GDP per capita in other affected countries. Consequently, the economic costs, measured in dollars, were relatively small for Ukraine. However, given diminishing marginal returns from money to welfare, the *welfare* loss would have been proportionately greater in Ukraine, as the poorer you are, the more a given loss of money reduces your welfare. Even though the economic costs of NotPetya, expressed in dollars, may have been lower in Ukraine than other countries, it seems plausible that the welfare costs were concentrated in Ukraine. The problem of using dollars as a proxy for welfare is particularly acute in this case, where there are large disparities in income across victims.[91]

For this reason, ideally, the social costs of events like NotPetya would be measured in terms of their effect on welfare, rather than their effect on money (Bronsteen et al 2013).[92] Quantifying

[89] “Various locations of the Beiersdorf pharmaceutical group were cut off from mail traffic for days. Beiersdorf said 35 million euros worth of second quarter sales were delayed to the third quarter” (Table 1).
[90] “According to Deputy Head of the Presidential Administration Dmytro Shymkiv, a former head of Microsoft's Ukrainian office, about 10% of all private, corporate, and government computers in the country failed that day. About half of them are beyond repair.” (Hromadske 2017)
[91] Note that this is a problem from the utilitarian point of view embodied by standard cost-benefit analysis, but also from other ethical points of view, such as egalitarianism and prioritarianism (Broome 2024).
[92] Similar problems have arisen in other domains. Early versions of the IPCC reports stated that the value of a statistical life in each country is proportional to income in that country, which implies that the value of a statistical life in poor countries is very much lower than in rich countries (Broome 2024). But lower

social costs in terms of money has the advantage of being widely popular and in some ways easier to calculate, but it is still subject to fundamental conceptual problems.

Nonetheless, in this report, we quantify costs in dollar terms that are roughly ‘welfare-adjusted’. 75% of infections were in Ukraine, which implies a 4X adjustment to the Crosignani et al estimates. However, the raw economic damages to Ukraine were lower, which suggests that 4X is an upper bound. We think a rough adjustment of 2X is justified, though we are very uncertain about this.

A.10. Offence-defence balance

The norm of openness in cybersecurity

Among key actors in cyberoffence and cyberdefence, there is disagreement about the merits of openness with respect to dual use cyber tools. Many people argue in favour of openness (e.g. Schneier 2004). Cyber defenders release dual use tools (e.g. Metasploit) all the time. For vulnerability disclosure, many key actors practice *coordinated disclosure*: if someone finds a vulnerability, they are expected to first inform the vendors in order to give vendors the time to patch the vulnerability, and then after a delay of a few months, publish the vulnerability publicly.

However, many non-malicious actors do not practice openness. Most importantly, state intelligence agencies often do not practice responsible disclosure, but instead use cyber tools in their cyberattacks, usually for espionage. The Shadow Brokers malware tools are one example of this. The NSA held on to these tools for at least five years without disclosing them publicly. The leak of the Shadow Brokers tools occurred three years after the introduction of the Vulnerabilities Equities Process under President Obama, under which vulnerabilities found by intelligence agencies were disclosed by default to the vendor (Healey 2016; Thompson 2021). Microsoft criticised the NSA for not disclosing the vulnerabilities sooner (Microsoft 2017).

When considering the merits of openness, it is important to distinguish: (1) whether openness would decrease the rate of cyberattacks; and (2) whether openness would increase social welfare. Openness may decrease the risk of cyberattacks, but it is an open question whether that would always produce better outcomes for the world. Obviously, many cyberattacks are socially costly. But state intelligence agencies value the ability to conduct cyberattacks against criminals, terrorists and rival states, and in some cases this may be socially valuable.

It is therefore unclear what our prior should be about the merits of openness for AI-cyber capabilities. This in part depends on broader worldview judgments which may differ across software vendors, civil society groups and state intelligence agencies.

willingness to pay to reduce the risk of death in poor countries reflects the fact that poor people have less money, not that their lives are worth less intrinsically.

Defining offence-defence balance

Offence-defence balance in cyber is usually defined in terms of the relative costs of attack and defence (Garfinkel and Dafoe (2019); Slayton (2017)). More precisely, this is given by the ratio of the defender's investment to the minimum offensive investment that would allow the attacker to secure some expected level of success (Garfinkel and Dafoe (2019), p. 251-252). A larger ratio corresponds to an easier attack. A technology *favours offence* if and only if it increases this ratio.

It is important to note that, on this definition, a technology might favour offence without increasing the number of expected attacks, and might favour defence without decreasing the number of expected attacks (Slayton (2017)). Whether an attack is likely to occur depends not only on the relative costs of attack and defence, but also on how attackers and defenders value their respective goals (Slayton (2017), p. 81). Even if attack is cheaper than defence at a given level of investment, attackers may simply value victory less than defenders, and so an attack might not take place.

For example, consider a scenario in which AI enables top tier state actors to find more elite vulnerabilities and exploits. Suppose that AI benefits attackers and defenders, but it is offence-favouring for worm attacks in that it increases the ratio of the defender's investment to the minimum offensive investment that would allow the attacker to successfully release a damaging worm. However, suppose also that attackers in this set are simply extremely unlikely to release a damaging worm because it does not further their goals. Since AI also benefits defenders by allowing them to more easily find and patch critical vulnerabilities, the overall effect of AI is to *reduce* the risk of worm attacks. This is true despite the fact that AI is, on the relative cost definition, offence-favouring with respect to worm attacks.

The likelihood of an attack also depends on the respective budgets of attackers and defenders. A defender may simply have more money than an attacker, which makes a successful attack unlikely, even if attack is cheaper than defence. Conversely, even if defence is cheaper than attack, attackers may have much larger budgets than defenders, so a successful attack may be likely.

Offence-defence balance is context-specific

Since the relative costs of attack and defence vary depending on context, offence-defence balance is not a property of cyberspace in general, but rather a property of relationships between particular defenders and attackers (Slayton (2017), p. 74). For example, the offence-defence balance for Chinese state espionage vs. the US is very different to the offence-defence balance for ransomware groups vs. specific businesses. For the same reasons, whether a technology (like AI) *favours offence* with respect to worm attacks depends on *exactly how* the technology increases exploit discovery.

Determinants of offence-defence balance

Offence-defence balance with respect to worm attacks depends on a number of uncertain factors (Lohn and Jackson (2022); Garfinkel and Dafoe (2019):

- **Elite vulnerability discovery rates over time**: the rate at which attackers and defenders find vulnerabilities over time, which depends in part on how fast there are diminishing returns from effort to elite vulnerabilities, and how many total elite vulnerabilities there are to discover.
- **Correlation between elite vulnerabilities discovered by attackers and defenders**: whether attackers and defenders tend to find the same vulnerabilities.
- **Elite exploit development time**: the gap between the discovery of vulnerabilities and the development of exploits for those vulnerabilities
- **Patch development time for elite vulnerabilities**: the gap between the discovery of vulnerabilities and the development of a patch.
- **Patch deployment time**: the gap between the development of a patch and the deployment of the patch by users.
- **The budgets of different attackers and defenders for elite vulnerability discovery and exploit development:** How likely an attack is to succeed depends on the total resources invested by different attackers and defenders.
- **Which specific agents gain access to elite vulnerabilities and exploits**: With respect to the cyber worm threat model, it is more concerning if lower skilled actors gain access to elite vulnerabilities and exploits.
- **The quality of firewalls, intrusion detection systems and endpoint detection and response**: Modern versions of these defensive systems can use machine learning to detect suspicious activity and prevent initial infection or widespread propagation, even if the malware exploits 0-day vulnerabilities.

There is a lack of good data on all of these parameters, so they are all very uncertain,[93] and the parameters interact in complex ways.

A.11. Defining different model release and safeguard policies

We consider three different policy approaches to model release and model safeguards:

1. P0: Open weight models.
2. P1: Proprietary models with refusals and anti-jailbreak measures
3. P2: Temporary protections with early access for defenders

[93] Lohn and Jackson (2022) estimate some related parameters. However, their data sources are relatively old, running up to around 2017, and it is likely that the trend has changed since then. The parameters may also be systematically different for the subset of vulnerabilities and exploits that we are interested in.

We now define these policies in more detail, as they were described to participants in our survey.

P0: Open-weight frontier models

By default, assume the following about frontier AI models:

- The **weights** of frontier AI models are **freely and publicly accessible** for anyone to modify and use. (However, in any evaluations requiring control groups without frontier AI access, those control groups are required not to use these models during the studies.)
- The scenario leaves it ambiguous about how other parts of the AI model are treated (such as whether the training code is also openly published). For a disambiguation of ‘open source’ as a term for AI see Seger et al. (2023).

Some points to consider as you form your forecasts:

- Experts believe that having access to the model weights makes it **meaningfully easier to circumvent any safeguards** the developer introduced, compared to accessing these via an API. A key difference between proprietary models and open-weight models is that the former is behind an API. Thus, if a **vulnerability is discovered**, it can be patched, and users are no longer given access to the older version. With open-weight models, new models that have these patches can be released, but the older versions cannot easily be ‘unpublished’.
- Such **vulnerabilities include** (but are not limited to):
    a. **Overcoming the model’s safety features**
        - AI companies train models to refuse to answer dangerous queries. However, with access to model weights, these safety features can be removed, e.g., through finetuning (i.e., giving the AI a few key examples where the ‘correct’ answer is to answer the question).
        - Although in some cases this can be done by accessing a model through a company’s API (see Qi et al. 2023), most frontier models don’t allow their latest models to be finetuned via their APIs or putting other restrictions in place (for example, see OpenAI’s policy, which currently allows finetuning on GPT-4o but not o1)
        - And even with an API that allows finetuning it will generally be easier to do this with unrestricted access to the model weights, as knowledge of the model’s architecture may make it easier to find vulnerabilities, and there is no risk of detection through company monitoring API usage.
        - Access to open-weight models has also allowed researchers to identify novel universal jailbreaks (see Zou et al. 2023) – i.e., carefully crafted questions such that the AI no longer recognizes that the developer intended for it to refuse these questions

        - Although jailbreaks now are much more sophisticated, to illustrate early examples include having users add “disregard all previous instructions” in front of the prompt (Russinovich et al., 2024)
    b. **Enhancing the model’s dangerous features.**
        - As well as removing safety training, with access to model weights, it is possible to enhance the dangerous capabilities of the model. For example, by:
        - ‘Recovering’ knowledge that the developer tried to get the model to unlearn. See Deeb & Roger (2024) as an example.
        - Fine-tuning the AI using data that may have originally been excluded from the training set, or proprietary data the actor has available, such as dual-use science articles
        - Importantly, such techniques do not necessarily have to be developed by people who do not intend to use the model to develop cyberweapons per se, but who try to increase an AI’s general capabilities and then share it via the Internet with its safeguards removed.
- However, you should also consider how AI being more open-weight might provide additional safety benefits that lower risk, specifically:
    - Open-weight AI may benefit defenders by helping them find and patch vulnerabilities.
    - Open-weight AI may allow a broader range of actors to identify vulnerabilities in AI models and ‘patches’ (NITA, 2024).
    - This might not impact the risk from these identified vulnerabilities (as the original ‘unpatched’ version of the model will remain available for threat actors to use), but this could be important for improving the security of future (more powerful) AI models when they are released.

Mitigation policies

We are also interested in your views on how the following mitigation policies could change the risk of large data-damaging worm attacks.

P1: Proprietary models with refusals and anti-jailbreak measures

The models used in the study (and similar models) are all proprietary and require users to access them via APIs that are subject to the safeguards described below [i–iii]. Companies train the models to refuse to respond to requests for potentially harmful information. Open-weight models are no better than the best open-weight models as of 31st August 2024.

- I.e. no 2026 open-weight model does meaningfully better than the 2024-July benchmark results from Meta’s Llama 3.1-401B s and would not create the same sized uplift described in the influenza RCT study

i. Before deployment, a pre-release test of 5 red-teamers working together full-time for 1 week can't identify a universal jailbreak, but 10 red-teamers working together full-time for 2 months are able to find at least one universal jailbreak. A universal jailbreak is defined as 'a type of vulnerability in AI systems that allows a user to consistently bypass the safety measures across a wide range of topics' and tested by whether a panel of cybersecurity experts the model can as a result accurately and answer in sufficient helpful detail a set of questions about vulnerability and discovery and exploit development.

- For comparison, a 2024 UK AISI evaluation found that "basic" jailbreak techniques ("either directly insert the question into a prompt template or follow a few-step procedure to generate question-specific prompts") caused current models to comply with 90-100% of harmful requests.

ii. After deployment, the companies developing the most powerful models have a voluntary goal of not letting any new universal jailbreak remain unpatched for more than 2-weeks over any given three-month period. To do so, each company has:

- A "bug bounty" programs that offer up to $15,000 rewards for anyone who identifies and reports a universal jailbreak for one of their models
- It would be similar to that currently run by Anthropic but all companies that have trained AI models similar to that in the scenario would have this program.
- For comparison, according to Zerodium, in general (non-AI) software, a zero-day vulnerability that allows you to bypass a phone's passcode or a PIN nowadays is worth up to $100,000 – and one that grants you zero-click remote code execution on Windows is worth up to $1,000,000
- 0.5 FTE (full-time equivalent staff members) who monitor the internet for mention of jailbreaks against their model and review instances flagged by automated processes (although it is left ambiguous how effective these are),
- If something is reported, they have 2 FTEs 'on call' who then spend up to 2-weeks of effort trying to patch it. If it takes more effort than that to fix it is left ambiguous how an AI company deals with it.
- For comparison, Google Project Zero (an elite zero-day finding group) reported that in 2021 they disclosed 63 critical security vulnerabilities that took the vendors an average of 52 days to fix, down from an average of 80 days 3 years ago. They have pushed for an industry standard of keeping this number below 90-days.

iii. The companies that own the proprietary models have information security practices at "Security Level 2" as described in the 2024 RAND report "Securing AI Model Weights: Preventing Theft and Misuse of Frontier Models" (see pp. 25-6). This security level is intended to describe "A system that can likely thwart most professional opportunistic efforts by attackers that execute moderate-effort or non-targeted attacks. This includes the operations of many professional individual hackers, as well as capable hacker groups when executing untargeted or lower-priority attacks." Security measures at this level include:

- Model weights are stored exclusively on servers (not on local devices, such as laptops) and are encrypted in storage with at least 256-bit strength encryption.
- The organization requires and enforces strong passwords, frequent software updates, and reporting of lost or stolen devices.
- A qualified security team is on call 24/7.

P2: Temporary protections with early access for defenders

The public release of the model has P1 level safeguards in place. However, a specific set of 'cyber defenders' is given access to a version of the model without any P1 cyber protections, i.e. with full vulnerability discovery and exploit development capabilities.

- The set of cyber defenders includes Microsoft, Meta, Apple, and Google
- It also includes the world's best highly vetted bug bounty hunters, including only:
    - Synack Red Team
        - Before joining the team, each prospective Synack Red Team member must first complete a 5-step vetting process that is designed to assess skill and trustworthiness (Pathways | Synack)
        - The Synack Red Team is made up of hundreds of the best pentesters and tech practitioners in the world, from countries across the globe (Synack)
    - Cobalt Core
        - Each of their pentesters has gone through a strict vetting process that only admits the top 5% of applicants (Cobalt.io)
        - Every tester is thoroughly vetted; the small percentage of applicants accepted onto the platform undergo ongoing peer review to guarantee high quality output (Cobalt Core: Become a Pentester)
    - HackerOne Clear
        - Eligibility includes Background checks, citizenship verification, $15K+ lifetime bug bounty requirement
        - ID verified, have exemplary platform performance and professionalism - agree to their Rules of Engagement (Careers | HackerOne)
        - Lifetime bug Bounty earned by participants : $15,000 (Careers | HackerOne)
        - Repeated criminal background checks.

After four months, the model is released under P0 security, i.e., is open weight.

A.12. For AI developers designing capability thresholds, it makes most sense to consider expected costs up to a period of around 6-12 months

Independent of the effect of AI on offence-defence balance, for misuse risk, there are also reasons for AI companies only to consider over 6-12 months when designing their capability thresholds and safety frameworks. There are two reasons for this. Firstly, open source models appear to be 6 months behind the current AI frontier (Epoch 2025). Since deployment safeguards to prevent cyber misuse of AI models can be easily removed from open source models, threat actors can easily switch to open source models without safeguards once these models are released.

Secondly, the leading extant approach in AI governance is to impose reporting requirements on models trained using a large amount of compute (e.g. >10^{25} or >10^{26} FLOP), which have tended to have the strongest capabilities. However, due to progress in compute efficiency and algorithms, the cost to achieve comparable performance is declining quickly over time. The gains in terms of effective compute from algorithmic efficiency are 2-6X per year (Epoch nd), and FLOP/$ is increasing by 1.6-2.9X per year (Epoch 2023). This means that some models trained on amounts of compute below the current threshold, and therefore outside regulatory requirements, will catch up to the current frontier in less than a year. Moreover, extant compute thresholds may be relaxed, or advanced models may be trained in countries that do not impose compute thresholds. As non-regulated models catch up, the benefits of imposing safeguards on larger models decline, as threat actors can simply switch to models without safeguards. Moreover, insofar as gains in future performance come from large amounts of inference compute, rather than compute for pre-training, fewer powerful models will be covered by current compute governance frameworks.