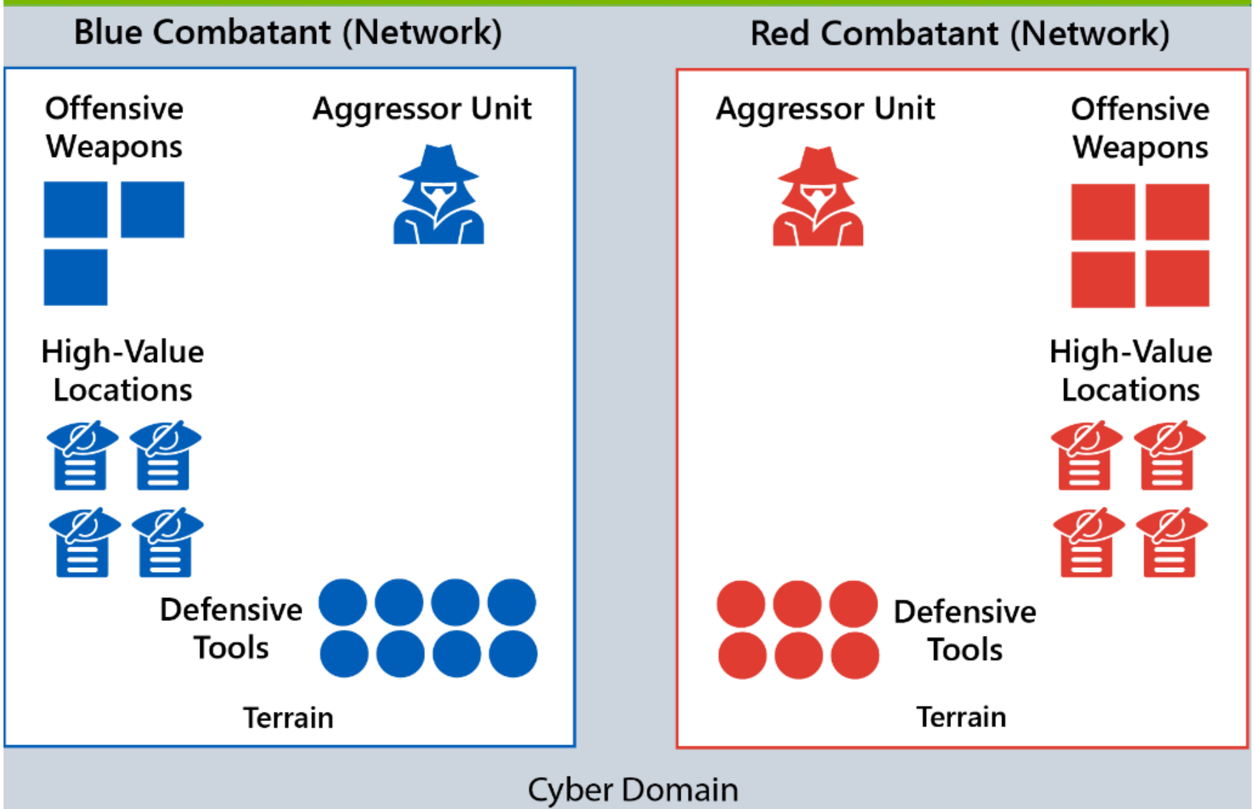
January 2024



A Novel Model of Cyber Combat

S. John Spey and Zeeve Rogozinski

DRM-2022-U-033642-1Rev

Abstract

In this CNA-initiated, Navy-funded study, we have developed a novel model for cyber combat that allows us to explore the complex decision space of cyber warfighting. Designed to be easy to modify and extend, the model uses abstract representations of elements of cyber combat. Using the model, we developed a statistical model of the interactions between network intruders and defenders and validated it with real-world data. Based on initial explorations with the model, we have learned that for a network intrusion to be successful, the intruder must be several orders of magnitude better at avoiding detection than the network defender is at detecting intruders. We have also found that in a competition between two equally skilled combatants, offensive cyber teams are best employed to support noncyber objectives and not on attrition of the opponent's capabilities in the cyber domain.

This document contains the best opinion of CNA at the time of issue. The views, opinions, and findings contained in this report should not be construed as representing the official position of the Department of the Navy.

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

Public Release 1/11/2024

This work was created in the performance of Federal Government Contract Number N00014-22-D-7001.

Cover image: CNA.

Approved by: **January 2024**

John J. Clifford

Dr. John J. Clifford
Director, Cyber, IT Systems, and Networks Program
Systems, Tactics, and Force Development

Request additional copies of this document through inquiries@cna.org.

Executive Summary

As part of a CNA-initiated, Navy-funded effort, we have developed a computer-based model of cyber combat at the high-tactical level or operational level. The model is designed to allow experimentation with longer term strategies across multiple attempted or successful network intrusions, so it has a high level of abstraction. Instead of modeling specific offensive weapons and defensive tools, it uses single values to summarize their effectiveness. Those values then determine the probabilities of potential outcomes of interactions between model participants.

A more specific example of the abstract nature of the model is that the progress of aggressors moving through target networks is modeled as a countdown timer that the aggressor must wait through before reaching an objective rather than the model checking repeatedly for how much incremental progress that the aggressor makes. From the high-tactical perspective, what is of interest is whether the aggressor can reach its target location inside the target network before being detected and ejected from the network.

The simplistic nature of the model allows it to run a complete model simulation in a fraction of a second. The model iterates the same simulation many hundreds or thousands of times with the same initial conditions and strategies to create a distribution of outcomes. The initial conditions can then be changed before iterating the model another several hundred or thousand times. This approach allows the model to be used to explore new scenarios rapidly. If the code of the model is also modified to implement different force commander strategies, the model can be used to explore an even wider range of scenarios.

Model description

In its current state, the model runs with two or more sides, called combatants. Combatants attempt to gain illicit access into each other's cyber terrain, seeking either to create effects in the cyber domain by neutralizing their enemy's defensive tools or offensive weapons or to gain access to high-value locations inside the enemy's network that provide benefits outside of the cyber domain. Because the model is only of the cyber domain, when aggressors reach locations that provide them with benefits outside cyber, the model rewards their combatant with abstract points. The model's output includes a distribution of tools and weapons remaining on each side and the points each combatant scored.

Validation

We have partially validated the model by comparing the model's results for how long network intrusions should last before they are detected with real-world data on the durations of network intrusions. Rather than directly simulating individual network intrusions, we analyzed both the model processes and real-world data to find the distribution of intrusion durations. We developed an underlying statistical model driving these distributions and found that both the model and real-world data predict the same distributions.

Insights

Using the model, we have also developed several insights about cyber combat via some simple explorations and analysis:

- For intrusions to last long enough for aggressors to accomplish any objectives, **aggressors must be much better than defenders** at the tactical level. If aggressors are not several orders of magnitude better at avoiding detection than defenders are at detecting them whenever the model tests for detection, network intrusions end very quickly.
- Assuming all defensive tools have an equal probability of detecting an aggressor inside the network, the marginal benefit from adding one more tool to the network decreases as a reciprocal of a root of the number of tools. This rapid decrease suggests **improving the detection probability of existing tools rather than adding another tool** equal in value to existing ones.
- All else being equal, **the best strategy is to go after only objectives that provide benefits outside the cyber domain**. It is a waste of effort for any aggressor units to go after the other goals of neutralizing the enemy's tools or weapons that are not yet in use.

Contents

This page intentionally left blank.

Cyber Warfighting Model

This report describes a model that simulates cyber conflicts at the high-tactical or operational level of war. The model's objective is to allow a cyber force commander with one or more tactical-level offensive aggressor[1] cyber units to experiment with different strategies and priorities. The model represents tactical details as simply as possible to avoid any dependence on details about individual networks, cyber defenses, and offensive cyber weapons.

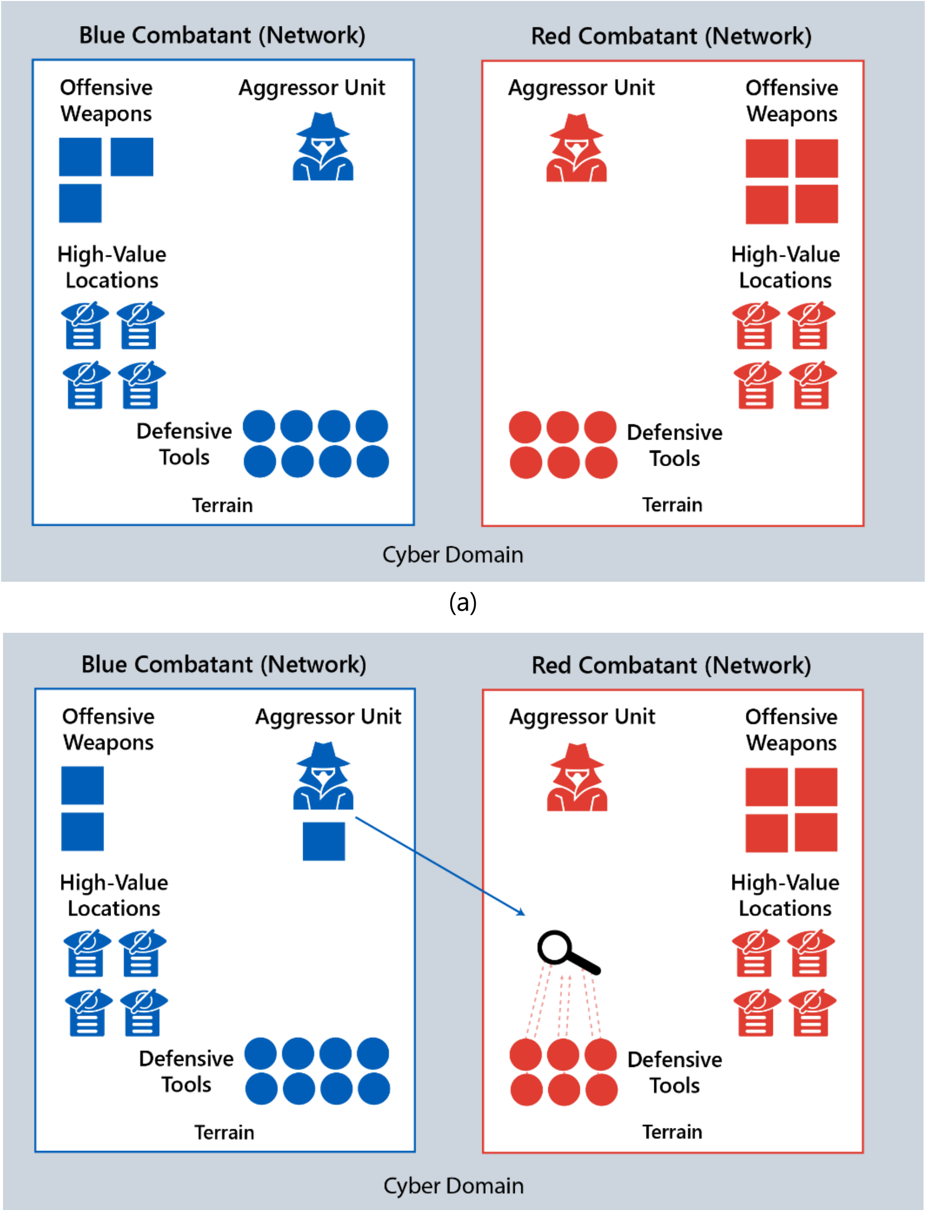
The simplistic nature of the model allows it to complete a single simulation in a fraction of a second. Many hundreds or thousands of iterations of the simulation with the same initial conditions and strategies can be done to create a distribution of outcomes. The initial conditions can then be changed before running the model another several hundred or thousand times. This approach allows the model to be used to explore new situations rapidly. The model code is written to be easily modified, allowing a wide range of scenarios and strategies to be explored.

In its current state, the model can simulate an arbitrary number of combatants. Combatants send **aggressor units** using **offensive weapons** into each other's **cyber terrain**, seeking to discover and neutralize their enemy's **defensive tools** or **offensive weapons**, or they seek out other high-value **locations** on an enemy's network to score abstract points. A single iteration of a given scenario runs until either one combatant accumulates a set number of points or a set number of timesteps have occurred.

Figure 1 depicts the components and basic actions of the model. It shows two combatants, Blue and Red, at the start of a model run in (a) and after a Blue aggressor unit has penetrated Red's network in (b). During each model run, each combatant's aggressor unit maneuvers into the other combatant's network and seeks out target locations determined by its commander. If an aggressor is detected by enemy defenders, those defenders may kick it out of the network, forcing it to start over with new offensive weapons.

[1] This report, and the model itself, uses the term *aggressor* to describe units that penetrate an enemy network and take illicit actions there because the terms *attack* and *attacker* have specific doctrinal meanings in cyberwarfare. Specifically, Joint Publication 3-12, *Cyberspace Operations* [1], defines a *cyberattack* as a cyberspace action that creates or leads to noticeable denial effects, which does not cover all illicit cyberspace activities inside an enemy's network.

Figure 1. Visual representation of the simulated environment inside the model



(a)

(b)

Source: CNA.
Note: Panel (a) shows the components of the model before any unit has begun aggressing. Panel (b) shows how one aggressor unit would penetrate the enemy's cyber terrain.

Model overview

Our model is focused on the decisions faced by the cyber force commander at the high-tactical or operational level.[2] The decisions of interest to the model are what priorities and objectives the commander should assign to their offensive and defensive forces and what strategies those forces should employ to achieve them. The ultimate goal of the model is to allow rapid experimentation and exploration of different priorities, objectives, and strategies set by the commander.

The model abstracts away the tactical-level details as much as possible to focus on decision-making above the low-tactical level of war. For example, although the details of the target network and how it is defended are of the utmost importance to the individual aggressors penetrating the network, what matters at the high-tactical level is whether the intrusion is successful and how long it takes to achieve its objectives.

The general premise of the model is that the combatants are sending aggressor cyber units against each other to achieve some goal in the cyber domain. This goal is assumed to support some higher level objectives, such as supporting a higher echelon or satisfying intelligence collection objectives.

The commander's decisions are represented in assigning one or more units their offensive weapons to use and one or more goals to achieve and defining a particular strategy for how they will execute their network intrusion. For aggressors with sufficient skill to spend time specifically reducing the chance that they will be detected inside an enemy network, the commander can also decide what level risk must be reduced to before they progress through the network. The commander may also decide some aspects of the defensive strategy for defending their own networks.

The model user sets the model configuration, which consists of defining the combatants and their initial state, such as number of aggressor units, number and quality of defensive tools and offensive weapons, and so on. It also involves defining basic strategy aspects that each combatant's commander directed their force to follow, such as when to neutralize an aggressor unit detected in their network and what objectives their aggressor units should have. Then the user starts the model run.

Interactions between the two combatants are resolved randomly, so model runs with the same configuration will produce different results. The states of each combatant in the model include how effective their offensive and defensive capabilities are. The relative skills of each side

[2] Identifying what constitutes operational-level cyberwarfare, as opposed to cyberwarfare at the high-tactical level, is beyond the scope of the current effort.

determine the probability of each potential outcome of each interaction. A random-number draw is then used to choose a particular outcome.

Once initiated, the model iterates hundreds or thousands of times without user intervention. A single iteration completes in a fraction of a second, so the model code is designed to run the model hundreds or thousands of times with the same configuration and then present the state of the combatants at the end of each iteration as its results.

Literature Review

We conducted a literature review of cyber modeling and found that the model presented here is indeed novel. Current cybersecurity models tend to focus on the behaviors of different types of threats and the vulnerabilities of a network or system connected to cyberspace. These models require intimate knowledge of their and their enemy's cyber domain, including the hardware and physical nature of the system that the network is monitoring (e.g., power plants, centrifuges). There is a plethora of models to choose from, each designed to simulate different functions of cyber [2-3]. Some of the more popular and well-developed cybersecurity models include Lockheed Martin's Cyber Kill Chain [4] and MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) model [5].

Lockheed Martin's Cyber Kill Chain [4] outlines seven steps that an aggressor will take when gaining illicit access into a system and models different courses of action that hackers may take at each step. This linear model is simple to understand and can easily highlight major vulnerabilities between each step; the Diamond model [6] was later developed to identify pivot points. A hacker will tend to follow the path of least resistance, and this model simulates all the possible trajectories to highlight where a hacker would most likely strike as they try to break into the system. MITRE's ATT&CK model [5] describes actions an intruder may take while already inside an adversary's domain. This model catalogs all possible tactics and techniques that a hacker may take at each stage and maps specific procedures to each step. One can easily customize unique pathways that an intruder may take and apply the necessary precautions to one's network. Threat Assessment and Remediation Analysis [7] is a methodology for identifying threats to a system and determining the appropriate countermeasures. This model catalogs and groups all possible intrusion vectors, outlines a series of steps that an aggressor takes during a cyber intrusion or attack, uses tools to match specific environments to potential intrusion vectors, and gauges the most likely possible vectors. All these models require relatively detailed information about a specific network to model a particular network intrusion.

Some models are designed to explore specific aspects of cybersecurity. For example, some models unify cyber and physical systems to better understand threats to physical systems that rely on cyber networks (e.g., water treatment systems, power plants) [8-9]. Some models instead focus on human error to try to improve system administrators' situational awareness to threats [10-14], whereas others try to hone modeling techniques to improve accuracy [15-17]. Regardless, all these cybersecurity models focus on specific actions taken by an adversary and the consequences of such threats on an unsecure system.

Relative to other cyber models, our model attempts to sit in the middle between highly abstract models, such as Lockheed Martin's Cyber Kill Chain or the general Advanced Persistent Threat model developed by Mandiant [18-19], and detailed simulations of specific networks or specific aspects of network security.

Model Description

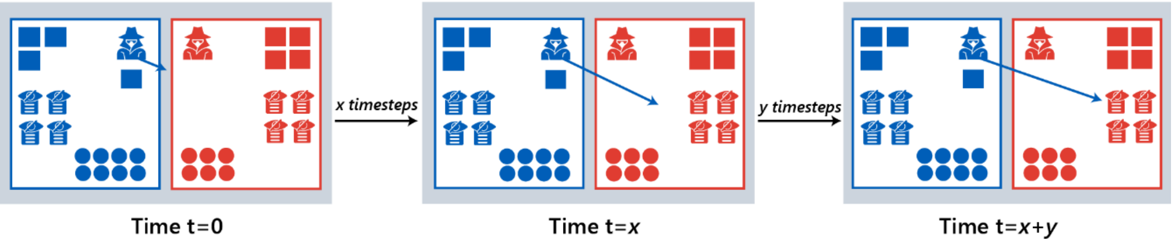
The model runs in timesteps, which represent an arbitrary but constant amount of real-world time. The action in the model is driven by aggressor cyber units. Each timestep of the model allows each aggressor unit to take one of the three following general actions:

- Waiting for a timer to fully decrement
- Actively mitigating the expected risk for taking an action
- Moving to another area in the battle space—into or out of a target network, to or away from a target location—which is modeled by changing the state that the aggressor unit is in

When a unit enters a new state, including at the start of the model run, it starts a countdown timer that counts in units of timesteps. Low-skill aggressors simply wait for the timer countdown to complete and then move to the next state; all the work necessary for them to progress to the next state is captured in the timer.

Consider an example shown in Figure 2. At the start of a run, a low-skill aggressor unit is put into the state of waiting in its own cyber terrain. The timer for this state represents the work by the unit to gain initial access to its targeted enemy network. When the timer expires, the unit is assumed to have gained access, and the model changes the unit's state to being inside the target network. When the unit enters this new state, it gains another timer that represents exploring and moving through the target network until the unit reaches its target location. A single model iteration continues executing timesteps until either one side earns a set number of points or a predetermined number of timesteps have occurred.

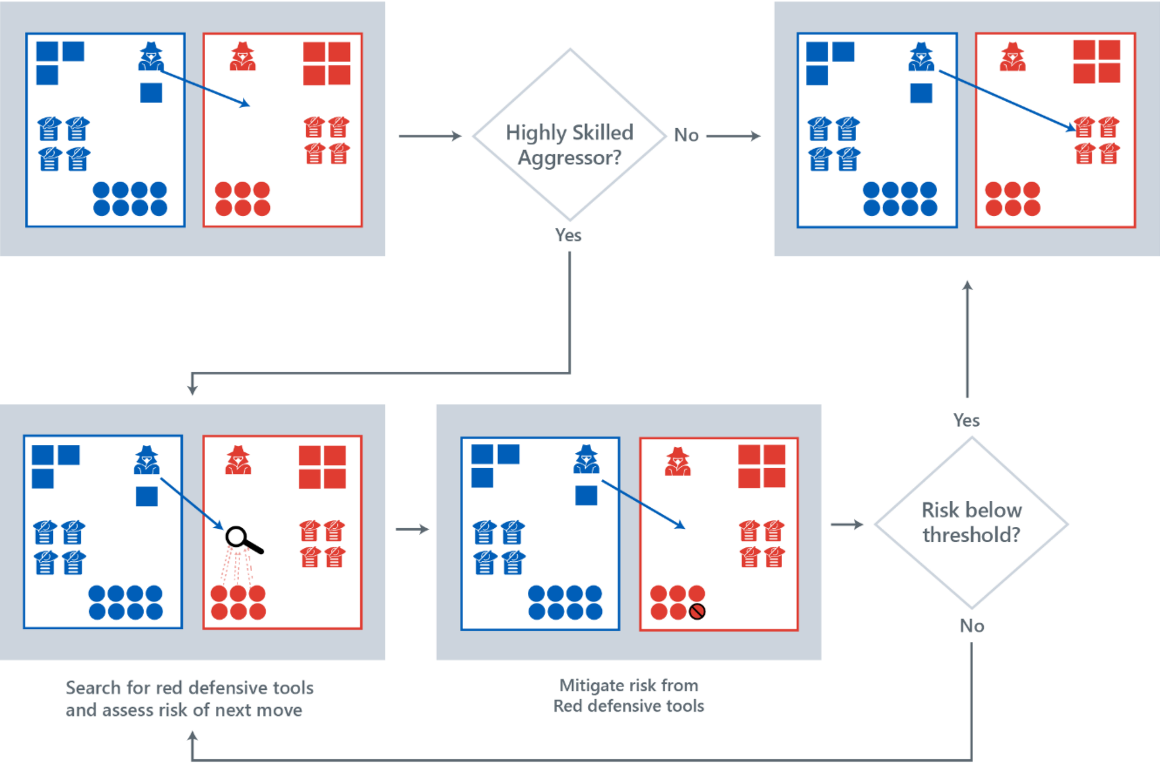
Figure 2. Basic timestep mechanics as Blue aggressor penetrates Red network



Source: CNA.
Note: At t=0, the Blue aggressor is attempting to penetrate Red's network. At t=*x* (after *x* timesteps), it is inside Red's network trying to reach a high-value location. At t=*x*+*y*, it is at the high-value location.

High-skill aggressors follow a similar process except they have an additional decision and action that represent them carefully assessing their risk of being detected by taking an action before taking that action. If the risk is too high, they first mitigate the risk and reassess it. If the risk is still too high, they mitigate it further. If not, they take the action. A comparison between the low- and high-skill aggressors' approaches is shown in Figure 3.

Figure 3. Different behaviors of low- and high-skill aggressors: Blue aggressor moving toward a high-value location in Red's network


Highly Skilled Aggressor?
No
Yes
Yes
Risk below threshold?
No
Search for red defensive tools and assess risk of next move
Mitigate risk from Red defensive tools

Source: CNA.

In the model, all actions taken by aggressors as they progress through a network toward their objective are represented by a decrementing timer. High-skill aggressors assess the risk of the action represented by decrementing the timer before they decrement it. Low-skill aggressors simply decrement the timer.

Aggressors are assumed to be sufficiently capable to reach whatever their target is as long as they are not detected. If they are detected, the defender can neutralize the weapon that the aggressor is currently using, which has the effect of kicking the aggressor unit out of the defender's network and forcing it to start over with a new intrusion using a new weapon. The model keeps a list of all available offensive weapons that each combatant has. When an aggressor unit needs a weapon, the model assigns it a weapon on the list.

Aggressor units risk detection constantly when they are in another combatant's cyber terrain. Each timestep, every defensive cyber tool has a chance of detecting any aggressor unit in the terrain that it protects. For low-skill aggressors, this chance is the same every timestep regardless of what action they are taking. The probability of low-skill aggressors being detected by a given tool is determined by only the effectiveness of a tool and how difficult it is to detect the offensive weapon.

When high-skill aggressors are taking action to reduce the probability of being detected when they progress through the network, their probability of being detected is reduced in both the future timestep when they progress and the current timestep while taking the action. This ability is what makes high-skill aggressors highly skilled.

High-skill aggressors decide to actively decrease their probability of detection in the future when they assess that the risk of progressing toward their objectives in the enemy network is too high. This active mitigation does not directly progress them toward their objectives, it only makes it less likely that they will be detected when they eventually do progress. This detection mitigation action can be thought of as "staying put" in the enemy network and reducing their signature when they attempt to move in the future.

An aggressor's probability of being detected in a timestep when it is taking the future detection mitigation action is also lower than if it were progressing through the enemy network. The model assumes that the mitigation activity is much harder to detect than progressing toward an objective. The nature of the activity—mitigating risk of detection—is intrinsically about avoiding detection; no useful action to mitigate the risk of detection in the future can present an equally large chance of being detected.

Aggressor units can also be detected if the weapon that they are using is neutralized by one of their target's aggressor units. That is, assume that a Blue aggressor unit is inside Red's cyber terrain and is using weapon "A." If one of Red's aggressor units in Blue's cyber terrain finds a copy of weapon A, Red can immediately neutralize that weapon: now that Red has a copy of it, it can detect the weapon and keep it from being used against them now and in the future. As mentioned previously, neutralizing a weapon has the effect of removing it from the model and kicking out any Blue aggressor unit using that weapon in Red's network. If three Blue aggressor units are using weapon A when Red neutralizes that weapon, all three Blue aggressor units will be kicked out of Red's network. Any Blue aggressor unit that attempts to penetrate Red's network using weapon A will be kicked out of the network as soon as it gets inside it.

Model iterations and runs

A model iteration involves instantiating the model with a given set of inputs, executing the model logic for multiple timesteps until one combatant achieves its objective and “wins” or some timestep limit is reached, and then saving the results. A model run consists of hundreds or thousands of iterations using the same inputs. One model run produces a distribution of outcomes.

A representative model iteration begins with two combatants, each with one aggressor unit. Each combatant starts with some number of weapons and tools. Each aggressor unit is assigned one weapon from its combatant to use. When the iteration starts, each aggressor unit begins decrementing a countdown timer. When the timer ends, the unit moves into its enemy’s cyber terrain and begins decrementing another timer. When that timer ends, the unit is moved to one of its target locations, which is simply a location inside the target network.

We use three categories of target locations as objectives: offensive weapons, defensive tools, and high-value locations. If the objective is the enemy’s offensive weapons or defensive tools, reaching the objective will neutralize one of them. A neutralized weapon or tool is considered no longer effective and removed from the model. If an aggressor unit attempts to use a neutralized weapon, once the unit penetrates an enemy’s network it is immediately kicked out. It is then given a new weapon and must start its intrusion effort over. If the objective is high-value locations, then the unit will earn points for its side for each timestep during which it is at the objective. Each high-value location contains a limited number of points, representing the maximum value a combatant can gain by having an aggressor unit reach it.

Once the weapon or tool is neutralized or all the points at that location have been extracted, the unit will begin searching for another target location. The unit is assigned another timer and when the timer is complete, the unit will be at another location. This process continues until a combatant accumulates enough points to win or a predetermined number of timesteps have occurred.

Abstraction

The model takes a highly abstracted view of offensive weapons and defensive tools. In the real world, different activities on a target network using a given weapon will be detectable by only certain types of tools. Malware executing on a target PC probably cannot be detected by a tool that monitors network traffic, for example. An aggressor can defeat a tool that scans activity logs for suspicious behavior by ensuring that its actions do not leave log entries. Addressing these types of specific interactions has led other cyber modeling efforts to attempt to model specific networks and aggressors, to enumerate all possible types of aggressor or defender actions, or to focus on a specific aspect of network defense.

The model described in this report takes a broader view. The network terrain, weapons, and tools in the model represent arbitrary versions, which adds uncertainty to specific interactions. In addition to uncertainty on whether a network defender using a tool that can detect a particular weapon being used in a particular way actually will detect it, the model has the uncertainty of whether a tool is capable of detecting a weapon at all. These uncertainties are combined into the overall probabilities of detection. Every time the model tests to see whether a tool detects an aggressor using a given weapon, the chance of a successful detection includes the chance that the tool simply cannot detect the weapon at all.

This situation is not as far removed from a real-world cyber intrusion as it may seem. In the real world, network defenders employ tools with little knowledge of what specific weapons aggressors will use against their network. Defenders know nothing about an aggressor's weapon before they detect it. They have no way of knowing whether their network will be compromised by aggressors using a weapon that they simply cannot detect. Conversely, aggressors rarely have perfect knowledge of the network that they are attempting to compromise, including what defensive tools its defenders might use. Taking this view, the model simulates the defender from the perspective of the aggressor and the aggressor from the perspective of the defender.

Model mechanics

The interactions between the different elements of the model that occur every timestep are governed by the model's mechanics. This section describes the mechanics of how aggressor units move through the modeled cyber environment and how the outcomes of interactions between aggressors and defenders are determined.

Modeled combatant forces

The model includes two or more similarly structured combatants. Each combatant's forces are assumed to include the following:

- One or more aggressor cyber units, either low or high skill, that execute operations to gain access to and maneuver in a target network
- A stockpile of offensive weapons
- The ability to generate and employ new offensive cyber weapons
- Network defenders that employ defensive cyber tools
- The ability to generate and deploy new defensive tools to the network

Each combatant also has a network of its own that it needs to defend. Every network contains the following:

- The stockpile of offensive weapons that either are currently in use or could be used in the future
- The defensive tools that potentially detect offensive weapons in the combatant's terrain
- One or more types of high-value information

These are each represented as potential goal locations for aggressor units. When a unit is aggressing, its tactical-level objective will be to reach one or more of these goal locations.

Weapons and tools potentially represent a wide range of items and properties. A weapon could be an advanced and complex piece of malware, or it could be one or two small, specialized malicious applications and a set of advanced tactics, techniques, and procedures (TTP). A tool could be a defensive network device such as a firewall or a regular process of collecting and reviewing logs. This uncertainty facilitates further abstraction in the model; if any weapon could reasonably be detected in some way and every tool could reasonably detect some malicious activity, they will work in the model.

Timer duration

The model uses simple countdown timers to represent an actor in the model performing an action that takes longer than one timestep. These actions include the following:

- Gaining initial access to another combatant's cyber terrain
- Reaching a target location
- Exploiting a target location
- Moving to another target location
- Developing a new tool or weapon

The durations are generated by drawing a value from a Poisson distribution with a mean equal to the expected average duration of each action. We use the Poisson distribution because it has several useful attributes. It has a single parameter that is equal to the mean, so time durations in the model can be defined using only their average duration. The distribution supports only integer values equal to or greater than zero, so it will produce only numbers that are valid starting values for the model's countdown timers. Finally, for double-digit or larger parameter values, the shape of the distribution is relatively symmetric about the mean and begins to approximate a discrete version of the normal distribution with a mean equal to its variance. This property ensures that the numbers that the distribution generates will be relatively close to the mean, with roughly half the values returned by a random draw being within 5 percent of the mean value.

The model takes an abstract view of all changes in probability. There is some daily variability in all activities. The people performing the activities slowly learn how to perform the activities better, and the environment gradually changes in ways that make their activities more difficult. If these variations are averaged over time, which the model does, the probabilities become effectively "memoryless" and can therefore be modeled via the Poisson distribution. The only changes in probability that the model considers significant are those that are already reflected in the neutralization of a tool or weapon.

New tools and weapons are created during a model run via abstracted processes represented by timers. Each combatant has one countdown timer for creating tools and another for creating weapons. When one of these countdown timers completes, the combatant gains either a tool or weapon, as appropriate.. Regardless of what an aggressor does to a target network, the network's defenders can procure new equipment and commercial software to deploy in the network's defense. New offensive and defensive software can be developed on an isolated external network or procured commercially.

Low- and high-skill aggressor units

The model uses the same mechanics for both low- and high-skill aggressor units. All aggressors have a maximum assessed risk that they are willing to accept for their next action. For low-skill aggressors, this risk level is represented as positive infinity. The actual assessed risk by a low-skill aggressor will always be equal to or below positive infinity, so low-skill aggressors never take actions to mitigate their risk. They never even actually assess risk, because when the risk of an action has not yet been assessed the model assigns it a value of positive infinity. As a result, as soon as a low-skill aggressor decrements a timer, it believes that the risk of the next decrementing, which it has not yet assessed, is not too high for it.

The key mechanical difference between high- and low-skill aggressors is that high-skill aggressors' acceptable risk level is smaller than positive infinity. Immediately after high-skill aggressors decrement a timer and before they can assess the risk of decrementing it again, the value of their perceived risk of decrementing it again is set to positive infinity. Unlike low-skill aggressors, high-skill aggressors consider this risk to be too high, so they start the assess risk → mitigate risk → assess risk cycle shown in Figure 3. How this detection probability reduction is implemented mechanically is explained in the next section.

Detection attempts

For each timestep that an aggressor is in the defender's network, the defender has a chance of detecting it. The base chance of detection is determined by the quality of the defensive tools that have not been neutralized by the aggressor. Each tool has a base chance of detecting an arbitrary aggressor. This chance is modified by dividing the tool's detection chance by a value representing how difficult a weapon is to detect. If the weapon is being used by a high-skill aggressor, the chance might be further modified because the aggressor either previously took a detection mitigation action or is currently taking one.. In both cases, the risk would be reduced by a factor set in the model. The resulting probability is the chance that the tool will detect the weapon.

To save computational time, the resulting probabilities of each tool are combined into one final probability before testing. Detection by one tool is treated the same as detection by another. So if there are multiple tools, there will be many possible outcomes that result in the aggressor being detected at least once but only one in which it is not, which would be when all defensive tools fail to detect the aggressor. We calculate the probability of this singular event of all tools failing to detect the weapons, which we call $p_{f,\forall}$, with the following equation:

$$p_{f,\forall} = \prod_{i=1}^{n} \left(1 - \frac{p_{d,i}}{w}\right) \qquad (1)$$

Here, n is the total number of defender tools; $p_{d,i}$ is the probability, ranging from 0 to 1, of tool i detecting any weapon; and w is the value representing how easy or difficult the weapon is to detect. If the aggressor is high skill and has reduced the chance of being detected on a given timestep, the model will implement this reduction by reducing w by some factor. If the term $^{p_{d,i}}/_{w}$ is larger than 1, the model sets it to 1 instead. The equation produces a probability from 0 to 1. This calculation is performed for each set of w and $p_{d,n}$ whenever detection is tested. For each timestep, the model compares the calculated $p_{f,\forall}$ to a random number drawn uniformly from the range (0,1] when testing for aggressor detection.[3] Assuming the aggressor does not choose to withdraw from the target network first, a given network intrusion will last until the random number drawn is greater than $p_{f,\forall}$.

Aggressor unit effects at objective locations

Once an aggressor unit reaches one of its objective locations, the effect it creates depends on the location. There are three location types: enemy weapons, enemy defensive tools, and high-value locations.

Enemy weapons

One objective location type is the enemy's offensive cyber weapons. This location represents the aggressor unit getting into the portion of the enemy's network from which the enemy launches its own offensive cyber operations. When a weapon is reached and the associated countdown timer ends, that enemy weapon is considered neutralized. Once neutralized, the weapon is removed from the model. If the enemy uses the weapon against the combatant

[3] The (0,1] mathematical notation denotes that the range is all numbers that are larger than 0 and smaller than or equal to 1.

whose unit neutralized it, that combatant will neutralize the weapon at the time during the intrusion that its strategy directs.

The definition of an offensive weapon is intentionally vague. A weapon may be a single piece of very complex and advanced malware that contains multiple exploits and other capabilities. It could be a combination of publicly known malware applications that target specific vulnerabilities, custom scripts, and TTP. It could even be access to a valid privileged user account. Regardless of what particular weapons in the model represent, they all have various unique signatures that a defender can detect, and they can all be neutralized by some set of defender actions.

Enemy defensive tools

Defensive tool locations represent the aggressor discovering a particular tool and learning enough about it to neutralize it completely. Once neutralized, a tool no longer has any chance of detecting an aggressor and is removed from the model. Any time necessary for aggressors to understand and neutralize the tool is included in the countdown timer that starts when they reach the location with the tool.

Like offensive weapons, defensive tools are also vaguely defined on purpose. One tool could be a firewall. Another could be a set of logging practices and analysis. A third could be an antivirus application. Instead of defining different categories of tools based on real-world defensive cyber measures and then determining how effective each category of tool is against different categories of weapons, the model simply treats specific tool–weapon pairings as one element of uncertainty. The only difference between different tools in the model is their base probability of detecting a weapon given one chance.

High-value locations

The third type of location an aggressor could reach is a high-value location on the enemy's network. Such a location may hold information of high intelligence value outside the cyber domain, allow control of some physical system, or provide some other benefit outside of cyber. The model represents this location as a state in which the aggressor gains some number of points at every timestep. Each location has a maximum number of points, which represents the total amount of value at that network location. The number of points provided each timestep and how many timesteps the location provides points for can be set in the model.

This approach could be used, for example, to represent causing damage in some way via the cyber domain, such as manipulating information used by the target, destroying information (as in a cyber-ransom attack), or reaching a cyber–physical interface and manipulating it to produce an effect in the physical world. For example, the aggressor could reach the control system of a water purification plant and manipulate it to produce undrinkable water. Any objective that involves the aggressor reaching some particular location or category of location and staying there long enough to take a particular action would fit.

Points have no meaning outside of the model. They are used here as a way of tracking progress. The model currently uses achieving 1,000 points as the victory condition to represent one combatant accomplishing some significant amount of its cyber objectives, but this number can be changed easily.

Model strategies

The goal of the model is to include different strategies by the combatants to experiment with their effect on final outcomes. In this section, we describe the strategy options currently in the model.

Targeting weapons or tools

One strategy currently implemented in the model is choosing which objectives are assigned to aggressor units. Aggressor objectives could include neutralizing the enemy's tools, its weapons, or both, in addition to accumulating points. This strategy is controlled by each combatant's "goals" variable in the input file. The only way to influence anything outside the cyber domain is by accumulating points, so that must be an objective of every combatant. If an aggressor has neutralized all an enemy's tools, then the enemy cannot practically detect it. It can move through the enemy's network freely, accumulating points without any danger of being detected and kicked out of the network until the enemy generates a new tool. If the aggressor neutralizes all the enemy's weapons, the enemy cannot accumulate points of its own until it generates a new weapon.

Post-detection actions

Detecting high-skill aggressors is generally considered to be difficult. One way to address this difficulty is for a combatant to observe an aggressor after detecting it, see what it does, and attempt to minimize the aggressor's overall impact. In the real world, if defenders detect that an aggressor is stealing sensitive information, they can work to minimize the damage caused by the aggressor having that information. Neutralized defensive tools might be replaced and stolen offensive weapons might not be used in the future against the enemy.

The opposite approach to observing detected aggressors is for defenders to neutralize them as soon as they are detected. In taking this approach, defenders assume that they will be able to detect aggressors in their terrain quickly and neutralize them before they achieve their objectives. Enemy aggressors might be detected with effective defensive tools or from an effective aggressor unit that targets the enemy's offensive weapons. Regardless of how aggressors are detected, this strategy involves neutralizing all enemy weapons as soon as they are discovered. Aggressors using neutralized weapons are ejected from the target network on the next model timestep. If detection happens faster than new tools are developed, the enemy is forced to culminate offensively; it will need an operational pause to build up offensive capability again.

Currently, the model tracks whether an aggressor unit has been detected by the defenders, who have a list of what states the aggressor unit must be in for the defender to neutralize them. For example, a defender that observes detected aggressors might not neutralize aggressor units that are in the state of searching for a high-value location but will neutralize those that are in the state of being at a high-value location. Conversely, a defender that neutralizes all detected aggressors immediately is represented as one that neutralizes aggressors in all states.

In addition, if a detected aggressor reaches a high-value location and begins collecting points, those points are reduced to a fraction of what the aggressor would receive if it had not been detected. This reduction represents the defender mitigating the impact outside of cyber of whatever the aggressor's activity at the high-value location represents.

Model Validation

To partially validate our model, we analyzed publicly available real-world data of network intrusions and compared the results to the behavior that the model produces. The model is not mature enough to simulate a specific network intrusion, but for some given scenarios it should produce behavior that varies in proportion with one or more inputs. If we have real-world data that vary in a similar manner, we can compare them to model-generated data. In the case at hand, we use a set of real-world network intrusion durations to validate how the model predicts that intrusion durations should be distributed.

There are large differences between different network intrusions. One constant aspect is that initially the aggressor has illicit access to the target network and later either leaves the network or is detected by the defender. Virtually all publicly known network intrusions are publicly known because the defender has detected the aggressor. Assuming the time when the aggressor first gained access can also be ascertained eventually, most publicly known network intrusions will have a start date and a date of first detection. If we have a large-enough dataset of real-world network intrusions, we can characterize the time from first intrusion to first detection statistically and then compare that statistical characterization to our model's predictions.

We now derive how we expect the time from initial intrusion to detection to vary as the relative abilities of a low-skill aggressor and a defender to avoid detection or detect the other, respectively, vary. We will then describe a real-world data source of network intrusions and characterize the distribution of times from initial intrusion to detection. We conclude this section by comparing the two distributions, showing that they are similar.

Model intrusion duration distribution

As shown previously, during each timestep in the model every aggressor inside a target network is tested to see whether it is detected by the defender. The two outcomes of interest are either one or more defender tools detect the aggressor or none of them do and the aggressor remains undetected. The probability that all the defensive tools fail a given test, represented as $p_{f,\forall}$, was shown in Equation 1.

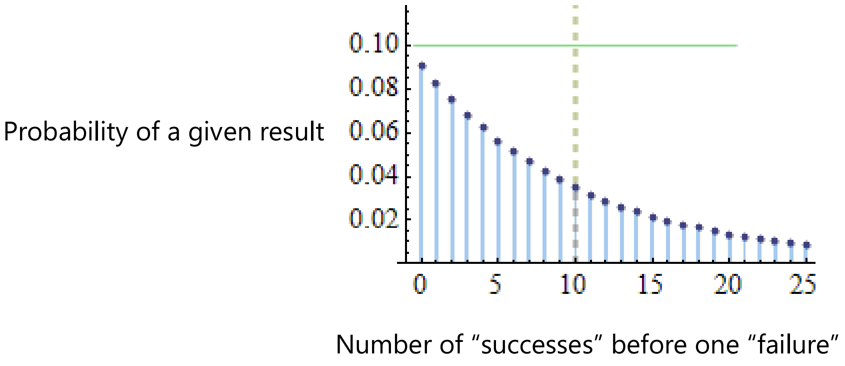
Each timestep, the model compares $p_{f,\forall}$ calculated according to Equation 1 to a random number drawn uniformly from the range [0,1]. Assuming the aggressor does not choose to withdraw from the target network first, a given network intrusion will last until the random number drawn is greater than $p_{f,\forall}$.

In the case of a low-skill aggressor in which $p_{f,\forall}$ is constant for a given network intrusion, a given intrusion can be treated as a repeated experiment consisting of a test with two possible outcomes—"success" and "failure"—each with a known probability of occurring that is repeated until the failure result occurs one or more times. This scenario is described statistically by the negative binomial distribution.

In our model, the experiment is when a low-skill aggressor is inside a target network, the test is the check performed every timestep to check whether an aggressor is detected, $p_{f,\forall}$ is the probability of the success result, and one failure result is sufficient to stop the experiment. For a given set of n, $p_{d,I}$, and w, we can use the negative binomial statistical distribution to predict how long one intrusion will likely last. We can also go further and calculate the probability mass function (PMF) for how long many such network intrusions with given values can be expected to last.

A representative negative binomial PMF for an experiment stopped after one failure is shown below in Figure 4. As the probability of a success result changes, the values on the horizontal axis change but the general shape of the curve does not.

Figure 4. Negative binomial distribution probability density function for an experiment that is stopped after one failure result



Source: Wikimedia Commons [20], with modifications by CNA.
Note: The probability of a successful result, denoted in the model description as $p_{f,\forall}$, is $^{10}/_{11}$. The horizontal green bar is the standard deviation, and the vertical dashed line is the mean.

The figure shows that for any given network intrusion, our model predicts that as long as a low-skill aggressor uses the same weapon and the defender uses the same tools every timestep, the most likely time an intrusion will be detected is as soon as it begins. This result may seem counterintuitive, but both the model itself and the negative binomial distribution suggest that it is correct. In each model timestep, the defender has some constant chance of detecting the aggressor. Looking at each timestep individually, the chances of the defender detecting the aggressor on timestep 10 are the same as on timestep 1. However, for the intrusion to last until timestep 10, the aggressor must first succeed in staying undetected for the previous 9 timesteps, and the chances of that happening are always less than one. The chance of the aggressor being detected on a specific timestep can be thought of as the chance of detection on any given timestep times the chance that the intrusion will last until that timestep. The chance of making it to any timestep beyond the first is always going to be less than one, so the chance of being detected on a later timestep will always be lower than that on an earlier timestep.

We note that the terms success and failure are somewhat arbitrary. In the case of the model, they are defined from the perspective of the aggressor, but the defender would likely switch the definitions. Consistent terminology does not appear to be used across various fields for the different properties of a negative binomial distribution in general.

Real-world intrusion duration data

To compare our negative binomial prediction for intrusion duration, we turned to the Vocabulary for Event Recording and Incident Sharing (VERIS) Community Database (VCDB) [21], a research repository of public network incident data. At the time of this analysis, the VCDB contained 8,868 total network incident records available in several standard data formats. The incidents are recorded using VERIS [22], whose format organizes incident data into standard fields. Fields that uniquely identify an incident in the VCDB are mandatory, and all others are optional.

Of interest to our research, one of the optional fields is the time from initial compromise to initial discovery. The data format for this field is such that submitters of intrusion data can use a unit of time of their choosing. Valid units are seconds, minutes, hours, days, weeks, months, and years. At the time of our research, 1,301 of the intrusions in the VCDB had a recorded intrusion duration in units of days or longer.

We note that because VCDB is a voluntary database, its data are vulnerable to at least two types of bias. One is that many organizations simply will not submit data on their intrusions to the VCDB, either because they do not know about them, do not have the resources to document them in VERIS, or simply do not want to share any information about intrusions into their network. The second type of bias is that real-world attackers are highly unlikely to submit their network intrusions to VERIS, biasing the data toward intrusions that are detected by defenders. Attempting to address the first type of bias is outside the scope of this report. The second type we consider of minimal consequence because our interest is specifically the time between when an intrusion starts and when it is detected and not intrusions that are never detected.

We analyzed how long intrusions lasted separately for records with units of days, weeks, months, and years and then fit a negative binomial distribution to the duration distributions for each unit of time. We separated incidents by the units used for duration because even though all durations could easily be converted into days, we found that incidents disproportionately lasted for integer amounts of units. It appears that when recording intrusion durations that lasted several weeks, months, or years, the data entry was rounded to the nearest integer value.

We show our results in the table and figure that follow. Table 1 shows information about the intrusion records that used each unit of time to report their duration. The amount of data is sizable, with the least commonly used unit of time, weeks, still having more than 100 records.

Table 1. Intrusion records using each unit of time to report duration

Unit	Records	Mode[a]	$p_{success}$	R^2
Days	203	0	0.75	0.95
Weeks	113	2	0.67	0.79
Months	663	2	0.77	0.83
Years	266	1	0.69	0.95

Source: CNA analysis of VCDB data.

Note: The table shows the number of records and mode of the data, the $p_{success}$ of the best negative binomial fit to the data, and the coefficient of determination, R^2, of the fit.

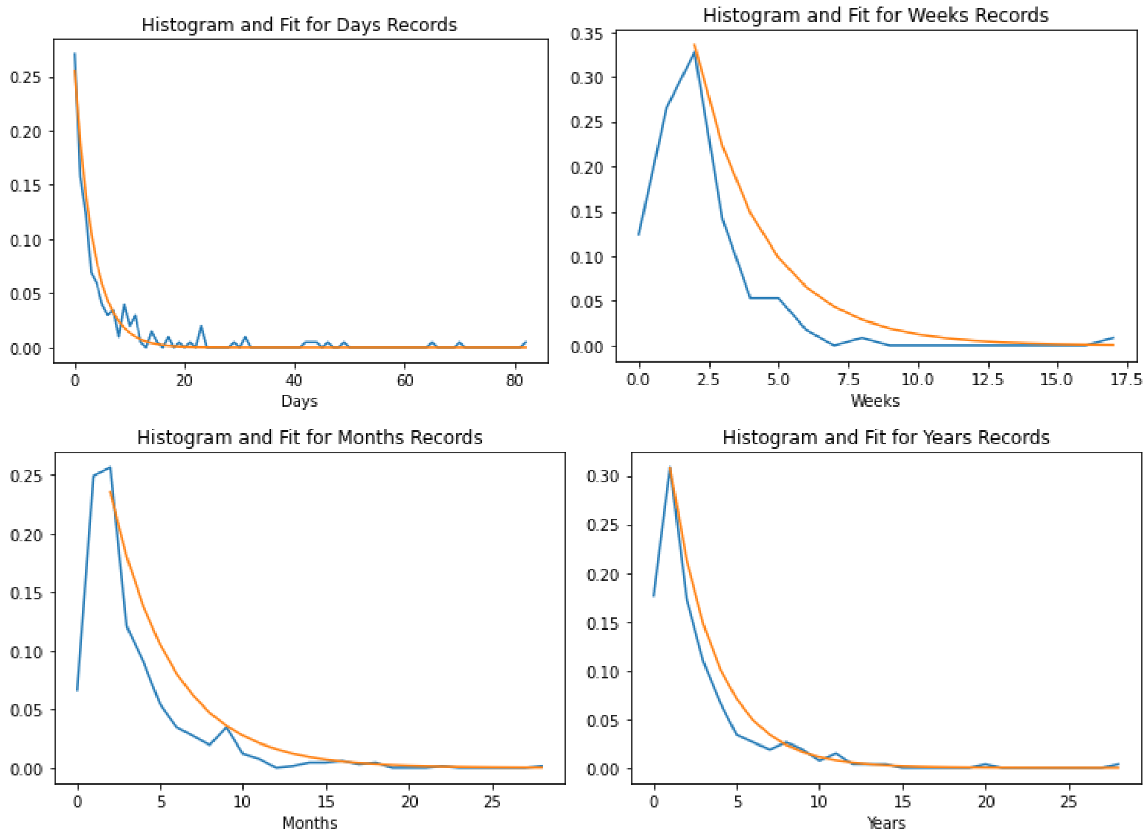
[a] Most frequently occurring value.

The most common value, or mode, is listed in Table 1 because the negative binomial PMF that stops after one failed test is constantly decreasing. Fitting the distribution to the data requires shifting it so that the distribution starts at the highest point in the histogram of the data. We assume that the data for units larger than "days" have modes greater than zero because shorter duration intrusions use a shorter unit of time. That is, it is likely that an intrusion with a duration of one week was recorded in the VCDB as lasting "seven days" and not "one week." We also show the probability of a successful test for the best negative binomial fit to the data. Labeled $p_{success}$ in the table, it is the same as $p_{f,\forall}$ in Equation 1.

From the R^2 (or coefficients of determination) values for each fit and the appearance of the four plots in Figure 5, we consider the negative binomial distribution behavior predicted by our model to be consistent with real-world network intrusions.

Although this validation assumed that the aggressors were low skill, high-skill aggressors could also produce an intrusion durations distribution similar to that generated by low-skill aggressors and found in the real-world data. High-skill aggressors consistently reduce the $p_{f,\forall}$ value by actively mitigating the risk of detection, so we cannot use the exact same approach as for low-skill aggressors, but if $p_{f,\forall}$ is always reduced to the same value, then the effect is similar to having $p_{f,\forall}$ have that value to begin with. If the risk of detection while taking mitigating actions is far enough below the high-skill aggressor's acceptable $p_{f,\forall}$, then all detections will still come while the aggressor is moving through the network. This has the effect of having an effectively constant $p_{f,\forall}$, which can still produce a negative binomial distribution.

Figure 5. Histogram and best-fit negative binomial PMF for real-world network intrusions by duration unit of time



Source: CNA, VCDB [21].

Note: Each chart shows histogram (blue line) and best-fit negative binomial PMF (orange line) for records of intrusions that denoted their duration in days, weeks, months, and years.

Model Insights

Although the primary objective of our current work has been to create the model, we have also performed some basic explorations with it that have led to certain insights, which we detail below.

Relative skill levels

As discussed in the previous section on model validation, most detected network intrusions are detected earlier rather than later because in the model, and likely in the real world, network defenders can constantly look for aggressors. To frame this scenario as a type of competition, the aggressor must defeat the defender constantly and repeatedly. If the defender wins once, the intrusion is over. Given that there is a random element in the detection test performed every timestep, we find that **the aggressor must be far better than the defender to have a long-lasting network intrusion**.

Figure 5 shows that many network intrusions last for several years. Assume that an aggressor is good enough to stay inside a target network for an average of two years without being detected, that defenders attempt to detect the aggressor with all their tools once every workday of the year, and that all tools have the same chance of detecting the aggressor. If there are 220 workdays in a year, the negative binomial distribution for that aggressor being detected by a defender must have a mean of 440. For our formulation of the negative binomial, the mean can be calculated with the following equation:

$$\frac{p_{f,\forall}}{1-p_{f,\forall}} \tag{2}$$

For a mean of 440, $p_{f,\forall}$ must be 0.99773. A frequentist interpretation of this value is that 99,773 times out of every 100,000 tests the aggressor will be undetected or that the defender will detect the aggressor only 227 times in 100,000 tests.

As shown in Equation 1, if all defender tools have the same chance of detection, then the value of $p_{f,\forall}$ is determined by three values: n, p, and w. The ratio of p to w, which is the ${}^{p_i}/_{w}$ term in Equation 1, represents how much better (or worse) the defender is than the aggressor. If this value is larger than 1, the defender is better. If it is smaller than 1, the aggressor is better. The further away the ratio is from 1, the larger the difference.

We chose 5 and 100 as reasonable lower and upper bounds, respectively, for the total number of tools that a defender has and calculated ${}^{p_i}/_{w}$ for a $p_{f,\forall}$ of 0.99773. For n of 5, ${}^{p_i}/_{w}$ is 4.54 ×

10^{-4}, and for n of 100, ${}^{p_i}/_{w}$ is 2.27×10^{-5}. Both values are several orders of magnitude smaller than 1, which we interpret to mean that at the finest grain level in the model, the aggressor is several orders of magnitude better than the defender. Whether this is achievable for aggressors following the low-skill approach or whether it requires the high-skill approach of carefully assessing and mitigating down risk before acting is a question to pose to practitioners.

Better tools or more tools?

The more tools already present on a defended network, the less benefit is provided by adding another tool, assuming all tools have the same chance of detection. The exact relative improvement depends on the chance of detection, represented by the ratio of p_d to w and the total number of tools n. We can calculate the benefit provided by another tool as follows.

If all tools have the same probability of detection, then $p_{d,i} = p_d$ for all i, which allows simplification of Equation 1 from a repeated product into an exponential with n defensive tools as follows:

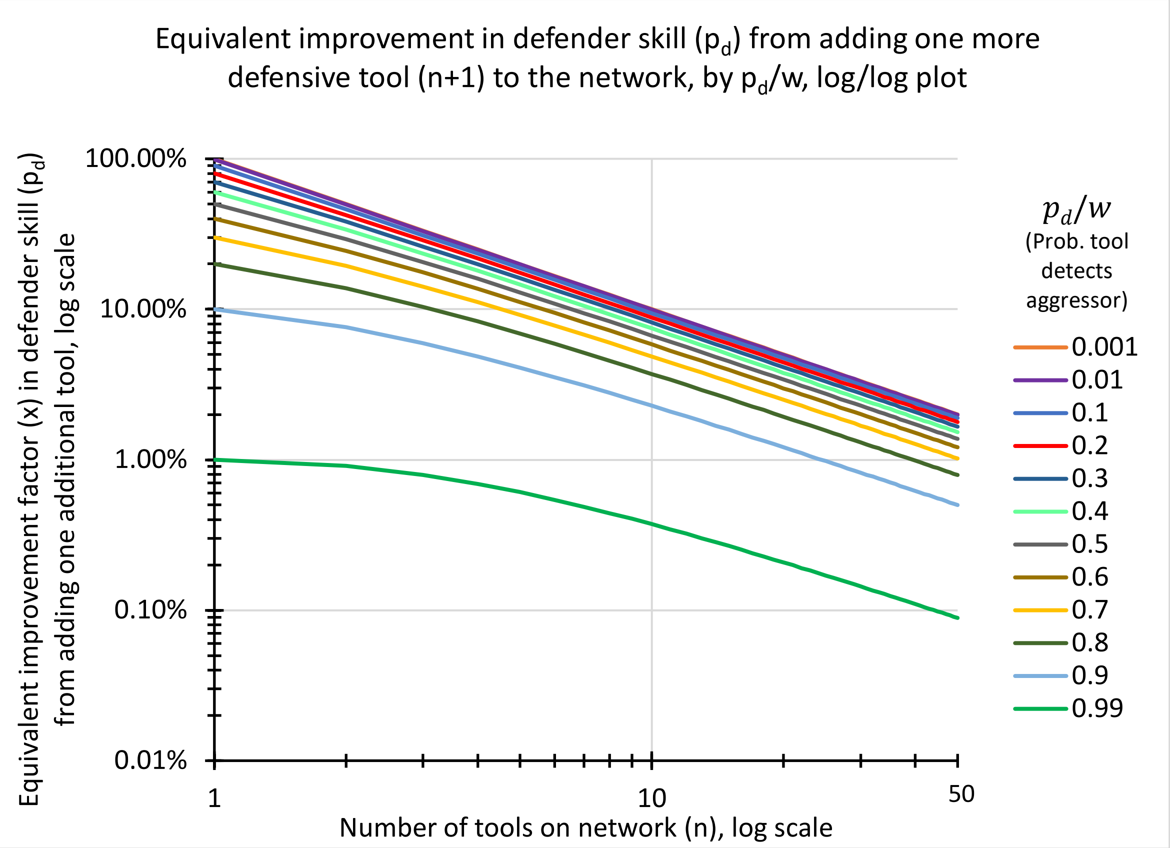
$$\prod_{i=1}^{n}\left(1-\frac{p_d}{w}\right) \equiv \left(1-\frac{p_d}{w}\right)^{n} \tag{3}$$

We are interested in relative improvement to cover as wide a range of real-world situations as possible. The increase in the chance of detection from adding one more p_d, done by increasing n by 1, has some equivalent increase in the value of p_d. That equivalent increase, denoted as x, would change p_d in Equation 3 into xp_d. The equivalence can be used to find x using the following equations:

$$\left(1-\frac{p_d}{w}\right)^{n+1} = \left(1-\frac{xp_d}{w}\right)^{n} \Rightarrow x = \frac{1-\sqrt[n]{(1-\frac{p_d}{w})^{n+1}}}{\frac{p_d}{w}} \tag{4}$$

Treating n as the independent variable, we plot the value of x that is equivalent to increasing n by 1 in Figure 6 on a log-log scale for a range of p_d/w values. The plot shows decreasing returns from adding another tool for all situations. If the probability of detection by a single tool is below roughly 50 percent, the decreasing returns are linear on a log-log plot, implying a power function relationship with a negative fractional exponent with x being a reciprocal of a root of n. Furthermore, the smaller the chance of detection by one tool, the less the detection chance influences the benefit of adding another tool. The lines for p_d/w values 0.1 and 0.01 almost overlap in the figure, and those of 0.01 and 0.001 do overlap.

Figure 6. Plot showing the equivalent increase in tool detection capability of adding another tool for a given p_d/w value



Source: CNA.

Making specific real-world recommendations based on Figure 6 can be difficult because unlike in the model, every network and defensive tool is unique. Certainly, if a network is barely defended, adding another tool will almost double the amount that it is defended—shown in the figure as the value for n = 1 approaching 100 percent on the vertical axis. Once a network has 10 tools, adding another of similar value will produce an equivalent increase of no more than 10 percent in all the current tools. The more tools defending a network, the more favorable it is to increase the chance of detection of existing tools, for example by providing additional training to tool operators or spending additional time configuring them or reviewing their output, rather than to add another tool. If a network has more tools than the existing defenders can manage, to the point where the tools might not be configured well, deployed everywhere they could be, or monitored, then reducing the number of tools in exchange for improved chances of detection by the remaining tools seems reasonable.

Tactics

When we have experimented with combatants having different goals, we have found that, all else being equal, **the best strategy is to go after points only**. It is a waste of effort for any aggressor units to go after the other goals of neutralizing the enemy's tools or its weapons that are not yet in use.

This finding is partly an artifact of the model design. Once earned, points are currently kept until the end of the iteration of the model without any risk of loss. Because victory is defined as collecting more points than the other combatants, it is somewhat obvious that prioritizing the collection of points is the best strategy.

Yet other aspects of cyber combat support a strategy of going after points only. In general, we restrict our range of scenarios to those in which aggressors are good enough to succeed with some regularity and note that the ability of aggressors to remain undetected dominates outcomes. Certainly, a scenario in which aggressors are rarely or never successful will predictably lead to defender victories. If aggressors are skilled, there is little reason for them to go after defensive tools. As discussed previously, Equation 1 is such that eliminating a single tool will not improve the chances of an aggressor succeeding by much unless the defender has very few tools in total. A 5 percent increase in a defender's chances of detecting an aggressor does little to address a difference of 4 to 5 orders of magnitude in skill between aggressor and defender. Neutralizing the enemy's weapons by finding them in the enemy's terrain also has an effect only if it can cause the enemy's aggressor units to start to run out of weapons. An enemy with two aggressor units and 3 weapons can create the same outcomes as one with two aggressor units and 50 weapons.

Future Work

The model described in this report could be used to explore a range of issues. In some cases, this exploration will require expanding the model to include additional concepts, and the model is designed internally to grow in this manner. We close the report by describing a few possible avenues of future investigation with the model.

Resource allocation

Real-world organizations must make decisions to optimize allocation of limited resources. As applied to cyber, offensive cyber forces could procure a small number of expensive but more capable tools or a larger number of less capable ones. They could organize into a small number of highly trained teams or a larger number of more poorly trained ones. Defenders likewise choose between spending resources on training personnel, procuring more tools, improving the configuration of existing tools, and other options.

These resource allocation decisions could be explored by generating various configurations for a combatant that represent various resource allocation outcomes and then running them in the model in different scenarios.

Aggressor tactics

The model could be used to explore a wide range of potential aggressor tactics. In what situations should a high-skill aggressor be especially cautious, and when should the aggressor instead accept more risk? Should a single aggressor team focus all its effort on a single intrusion with a single objective, or should it split its effort across several? If there are both easier-to-reach objectives with a smaller payoff and harder-to-reach ones with higher payoffs, what specific differences determine which are better targets for a given situation? How many different weapons should a combatant have available for its aggressor teams to use at any given time if it wants the teams to always have some capabilities to penetrate an enemy's network? Of the total weapons, how many should be in use at one time?

Defensive tactics

The model does not simulate specific defensive actions or details of the network. By adding them individually to the model, a combatant can estimate what benefit they might be providing.

For example, the network might be redesigned to have a chokepoint between all external access points and internal servers. The network might be split into several different segments, each of which requires different authentication. One network might be split into two or more separate and disconnected networks.

Modeling real-world activity

If data on a real-world combatant are available, it can be simulated in the model. Instead of the hypothetical, specific decisions and unknowns can be tried and run in the model to see the results both during more deliberate planning and, because of the speed with which the model runs, during an intrusion. Both aggressor and defender could explore different strategies in the model while fighting each other in real life.

The model can also be used to find unknown values of a given combatant. If a defender knows how often it detects aggressors in its network and can estimate the skill of the aggressors based on public or private information, it can estimate how good it is at detecting aggressors in its network. Previous interactions between two combatants can be modeled to develop expected behavior by both sides, such as how often one side expects to detect the other. These expectations can be combined with observations of one combatant by the other to estimate ground truth. For example, if one combatant's network penetration is lasting much longer than expected, perhaps the defender has detected it in a manner that the aggressor cannot observe. Or, if a defender suspects that aggressors are trying to get into its network but has not detected any network intrusion attacks, perhaps the defender should assume that its network has been compromised in ways that it has not detected.

Figures

References

[1] Joint Publication 3-12. 2018. *Cyberspace Operations*.

[2] Bodeau, Deborah J., Catherine D. McCollum, and David B. Fox. 2018. *Cyber Threat Modeling: Survey, Assessment, and Representative Framework.* MITRE. https://www.mitre.org/news-insights/publication/cyber-threat-modeling-survey-assessment-and-representative-framework.

[3] Couretas, Jerry M. 2018. *An Introduction to Cyber Modeling and Simulation*. Newark, NJ: John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/cnacorp/detail.action?docID=5520841.

[4] Hutchins, Eric, Michael Cloppert, and Rohan Amin. 2011. "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains." In *Leading Issues in Information Warfare & Security Research*. Edited by Julie Ryan. Reading, United Kingdom: Academic Publishing International Limited, 80-106.

[5] Strom, Blake E., Josheph A. Battaglia, Michael S. Kemmerer, William Kupersanin, Douglas P. Miller, Craig Wampler, Sean M. Whitley, and Ross D. Wolf. 2017. *Finding Cyber Threats with ATT&CK-Based Analytics.* MITRE. MTR170202. https://www.mitre.org/news-insights/publication/finding-cyber-threats-attck-based-analytics.

[6] Caltagirone, Sergio, Andrew Pendergast, and Christopher Betz. 2013. *The Diamond Model of Intrusion Analysis.* Center for Cyber Intelligence Analysis and Threat Research. https://apps.dtic.mil/sti/citations/ADA586960.

[7] Wynn, Jackson E. 2014. *Threat Assessment and Remediation Analysis (TARA).* MITRE. https://apps.dtic.mil/sti/citations/AD1016629.

[8] Adepu, Sridhar, and Aditya Mathur. 2016. "Generalized Attacker and Attack Models for Cyber Physical Systems." In *Proceedings of the Institute of Electrical and Electronics Engineers, Atlanta, GA, June 10-14*, 283-292. https://search.proquest.com/conference-papers-proceedings/generalized-attacker-attack-models-cyber-physical/docview/1814246665/se-2?accountid=10186.

[9] Zografopoulos, Ioannis, Juan Ospina, XiaoRui Liu, and Charalambos Konstantinou. 2021. "Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies." *IEEE Access*. doi: 10.1109/ACCESS.2021.3058403. http://arxiv.org/abs/2101.10198.

[10] Dutt, Varun, Young-Suk Ahn, and Cleotilde Gonzalez. 2011. "Cyber Situation Awareness: Modeling the Security Analyst in a Cyber-Attack Scenario Through Instance-Based Learning." In *Proceedings of Data and Applications Security and Privacy XXV, Richmond, VA, July 11-13*, 280-292. doi: 10.1007/978-3-642-22348-8_24.

[11] Karaarslan, Huseyin. 2017. “A Cyber Situational Awareness Model for Network Administrators.” Master's thesis, Naval Postgraduate School. http://hdl.handle.net/10945/52997.

[12] Moore, Andrew P., and Allen D. Householder. 2019. *Multi-Method Modeling and Analysis of the Cybersecurity Vulnerability Management Ecosystem.* Carnegie Mellon University, Software Engineering Institute. https://insights.sei.cmu.edu/documents/581/2019_019_001_550437.pdf.

[13] Swiatocha, Travis L. 2018. “Attack Graphs for Modeling and Simulating Sophisticated Cyber Attack.” Master's thesis, Naval Postgraduate School. https://apps.dtic.mil/sti/trecms/pdf/AD1060078.pdf.

[14] Szymanski, Bolelaw, Shivkumar Kalyanaraman, Biplab Sikdar, and Christopher Carothers. 2005. *Scalable Online Network Modeling and Simulation.* Rensselaer Polytechnic Institute. AFRL-IF-RS-TR-2005-291. https://apps.dtic.mil/sti/tr/pdf/ADA437818.pdf.

[15] Hassell, Suzanne, Paul Beraud, Alen Cruz, Gangadhar Ganga, Steve Martin, Justin Toennies, Pablo Vazquez, Gary Wright, Daniel Gomez, Frank Pietryka, Niraj Srivastava, Travis Hester, David Hyde, and Brian Mastropietro. 2012. “Evaluating Network Cyber Resiliency Methods Using Cyber Threat, Vulnerability and Defense Modeling and Simulation." In *Proceedings of MILCOM 2012 IEEE Military Communications Conference, Orlando, FL, Oct. 29-Nov. 1*, 1-6. doi: 10.1109/MILCOM.2012.6415565.

[16] Jajodia, Sushil, and Steven Noel. 2010. *Advanced Cyber Attack Modeling Analysis and Visualization.* Air Force Research Laboratory, Rome Research Site. AFRL-RI-RS-TR-2010-078. https://apps.dtic.mil/sti/pdfs/ADA516716.pdf.

[17] Ostler, Ryan. 2011. “Defensive Cyber Battle Damage Assessment Through Attack Methodology Modeling.” Master's thesis, Air Force Institute of Technology. https://apps.dtic.mil/sti/citations/ADA539975.

[18] “Targeted Attack Lifecycle.” Mandiant. Accessed Sept. 8, 2022. https://www.mandiant.com/resources/insights/targeted-attack-lifecycle.

[19] Mandiant. 2013. *APT1: Exposing One of China's Cyber Espionage Units.* https://www.mandiant.com/sites/default/files/2021-09/mandiant-apt1-report.pdf.

[20] “Probability Mass Function of a Negative Binomial Distribution.” Wikimedia Commons. Sept. 11, 2020. https://commons.wikimedia.org/w/index.php?title=File:Negbinomial.gif&oldid=454679749.

[21] VERIS Community. Apr. 20, 2021. Database. *Network Intrusion and Data Breach*. Online. https://github.com/vz-risk/VCDB.

[22] “The VERIS Framework.” April 2021. http://veriscommunity.net/index.html.

This page intentionally left blank.

This report was written by CNA's Systems, Tactics, and Force Development Division (STF).

STF focuses on systems and platforms at the tactical level of warfare, providing classical warfare analyses to help the US Navy and Department of Defense win the great power competition while meeting other warfighting requirements to simultaneously deter and defeat lesser threats. The division's mission includes analyzing and assessing alternative combinations of networks, sensors, weapons, and platforms to provide maritime warfighting capabilities in all warfare areas under realistic employment conditions for current operations and future force architectures.

This report may contain hyperlinks to websites and servers maintained by third parties. CNA does not control, evaluate, endorse, or guarantee content found in those sites. We do not assume any responsibility or liability for the actions, products, services, and content of those sites or the parties that operate them.

CNA

Dedicated to the Safety and Security of the Nation

CNA is a not-for-profit research organization that serves the public interest by providing in-depth analysis and result-oriented solutions to help government leaders choose the best course of action in setting policy and managing operations.

DRM-2022-U-033642-1Rev

3003 Washington Boulevard, Arlington, VA 22201
www.cna.org 703-824-2000