Multifactor Violator Model. Condition-factors and Resource-factors

O. Kireienko

***Abstract*—Attack scenarios with limitations were investigated. Resource-factors and condition-factors were set as two types of limitations. Resource-factors are spent at each step of attack and can be replenished completely or partially if a given attack step was successful. A situation, where successful completion of current step with one or more preceding ones is required to replenish resource-factors, is possible. After each step of attack the violator can "exchange" resource-factors to accumulate the required amount of those factors for the next step. The lack of the required amount of resource-factors may either forcefully interrupt an attack or to lower success probability or reduce the time required by protection side to discover the consequences of an attack. This article doesn't consider the change of relative cost of resource-factors, that is caused by urgency, so that all resource-factors have fixed cost regardless of violator's reserve of these resource-factors.**

Conditions-factors are fixed limitations for conducting an attack. Discrepancy of condition-factors makes it impossible to either start an attack or to finish the current attack's step. In certain cases the lack in one condition-factor can be compensated with excess of another condition-factor or via spending additional resource-factors.

The influence on resource-factors and condition-factors is laid as a basis of protection strategies. The strategy of increasing the values of condition-factors for violator decreases the total amount of attacks on a system by screening beginner violators. The threat level from groups of violators and from experienced violators will remain unchanged. The strategy of increasing the rate of resource-factors spending is designed to interrupt attacks in progress. Strategy of decreasing the amount of resource-factors that can be replenished after successful completion of certain steps of attack scenario is meant to decrease violator's interest in attacking specifically our system and to decrease the chances of attack repetition if an attack occurred.

***Index terms*—condition-factors; multifactor violator model; scenario violator model; resource-factors.**

I. Introduction

According to RD TPI 1.1-003-99

User violator model is an abstract formalized or non-formalized description of a violator.

The protection side develops the violator model while taking the specifics of the system that requires protection into account. The violator model is used to estimate threat level, pinpointing weaknesses in the system, prognostication of attack (or sequence of attacks), and for developing the system recovery scenarios should a successful attack occur.

Violator models can be divided into 3 categories according to their level of detail and purpose. These are informative models, script/scenario models and mathematical models of influence/impact/effect [1].

Informative models can only be used to get qualitative estimates of violator's abilities and as auxiliary materials for experts as they emphasize aspects that are crucial specifically for our information system.

Mathematical models of influence are used for quantitative estimates. Mathematical models of influence are *always* designed for a specific system because quantitative estimates of losses and system recovery time can't be derived solely from information about violator. A specific violator may pose a severe threat for your personal but at the same time he has no influence on network hardware and mobile devices.

The model that is described in this article is a scenario model according to the classification in [1]. Scenario models are used for prognostication of possible variants of system' behavior in response to conducted attack. Even with incomplete information about the system that requires protection, protection side can make assumptions about probabilities and/or frequencies of attacks and assumptions about possibility of their repetition since these values depend not only on hardware/software, network topology, personnel/staff education but also from specific tasks that are assigned to/being solved by our system. Attendance of the site in Internet depends on its content and date. Sites dedicated to External Independent Evaluation or similar exams in other countries at the end of academic year and sites of universities during the period of admission to said universities serve as a great example for the aforementioned statement. An increase of network traffic and malicious actions directed at the system itself or at the information within the system is bound to a specific date regardless of system's architecture and protection mechanisms that have already been implemented. Certain scenarios can be bound not to a specific date but to an event. The probability of certain attacks increases during the periods of system's update and maintenance while other attacks become temporarily impossible at the same time.

Scenario models help the protection side to solve the following problems:

1) to limit the amount of threatening scenarios, 2) to prevent one scenario from being amplified by another one (when an attack creates a vulnerability that can be used by other attacks), 3) to shift scenarios to the most favorable time period for our system (maximum amount of experienced staff members, fresh version of backup data, etc.)

Published on March 14, 2020.
O. Kireienko, National Technical University of Ukraine, Ukraine.
(e-mail: kirealex12@gmail.com)

II. LITERATURE REVIEW AND PROBLEM STAITMENT

Attack scenario is composed of a sequence of actions, that must be performed by a violator. At each step the violator chooses a possible variant of behaviour(e.g. different methods of reconnaissance or different methods of replacing traces after the attack). The choice of a certain variant depends on factor values that characterize the violator.

Reference [2] depicts threat scenarios for objects of critical infrastructure. It is worth mentioning that a certain scenario will be completed successfully if and only if all steps of that scenario are successfully completed. The quantitative estimates will account for importance/weight of each step.

Reference [3] depicts scenarios with subtargets. Availability of subtargets allows violator achieving his goal via different routes (while using different sets of factors) and performing additional actions or skipping optional ones in order to make an attack more efficient. Adding subtargets to the scenario significantly complicates attack prognostication. This problem is due to heterogeneity of links between subtargets and primary targets. Within one information system the scale and type of effect from compromising the subtarget may vary for different scenarios. For example, shutting down a server that contains a backup copy of information from our system in one scenario will only result in our inability to refresh the backup copy while in a different scenario it will prevent restoring the system from that backup copy.

Reference [4] claims that the amount of available scenarios for the violator is determined by his experience (a more experienced violator can recreate/mimic actions of the beginner but not vice versa). Although there are reasons to believe that the violator will abandon economically pointless attacks and will modify existing attacks to use the available resources more efficiently as his experience accumulates.

Reference [5] presents a direct (by looking through low level actions of all scenarios), revers/inverse (designing an optimal scenario from the end) and combined methods of scenario generation.

Reference[6] views scenarios where the violator may have accomplices even if those accomplices are heterogeneous (insiders and external violators, violators with different level of knowledge and assets, etc.), but the violator is forced to attack the most vulnerable fragment of the system. That contradicts conditions from other models. Even the protection side and system developer are often not completely aware about all vulnerabilities that are present in the system. Violator won't attack the weakest fragment of the system, he will attack **any** fragment of the system that has vulnerability. If the reasoning in [6] was correct there would be either 1) a single attack scenario (that affects the most vulnerable fragment of the system), or 2) a sorted set of attack scenarios (so that a protection side will be certain about the next attack if the current attack fails).

All harmful actions are divided into 3 categories in [7]. These categories are: 1) accidental actions, 2) deliberate actions with intent to harm the system and 3) deliberate actions without such intent. Switching between those categories isn't considered in [7] (e.g. we don't consider situation when a legit user makes a mistake while interacting with the system that causes the decrease in its security level and immediately decides to make use of the situation).

Three theories of information systems are depicted in [8]. These theories are 1) a theory of planned behavior, 2) a deterrence theory, 3) a protection motivation theory. Deterrence theory is particularly interesting in the scope of this work. This theory isn't restricted solely for/to intrusion detection systems (IDS). Within multifactor model affecting the resource-factors and condition-factors will deter violator from performing malicious actions in the system, although the magnitude of influence will depend on the selected protection strategy.

According to reference [9] the main components of deterrence are 1) the intensity and versatility of sanctions, 2) potential violator's awareness about those sanctions and 3) the possibility of detection of violation by the protection side. It is worth mentioning that the possibility of automation of aforementioned sanctions isn't considered in [9]. Thereby the decision about the type and magnitude of punishment for the violator is made by protection side after investigating the incident (the magnitude of punishment isn't equal to the losses, penalties may be higher and they can be supplemented with informal sanctions). However there is an exception. In the context of *Deception technique* violator's wasted time and resources for the failed attack are considered to be a punishment (or part of the punishment) for the violator.

The problem of deterrence theory is considered in general in [10]. This means that the type of sanctions may be different in nature from the type of attack as long as the condition of proportionality holds. On a global scale this means that one country may impose economic sanctions or even declare a war in response to the cyber-attack. On a smaller scale the type of response *must be* different. The protection side should not conduct a cyber-attack in response because it is illegal in multiple countries. Even if the protection side has an opportunity to perform such attack and avoid any responsibility, the problem of target isolation hasn't been solved yet (can we guarantee that only a violator will suffer from our attack?). The problem of deterrence from attack repetition wasn't considered to a sufficient degree either.

Reference [11] describes a category of attacks that can't be prevented with currently implemented protection mechanisms. This is a crucial difference because according to [11] extra protection mechanisms must be implemented to negate such attacks (one can't just reconfigure existing protection mechanisms). And the decision about categorizing an attack is made by protection side. It means that certain attacks will be mistakenly categorized due to insufficient competence of protection side (when it is possible to protect the system from an attack but the protection side doesn't use available protection mechanism to their full extent; e.g. a PC user who always interacts with the system from the administrator account or root-account)

or due to the conflict of protection mechanisms and application software(when it is possible to protect against certain attack but it will create a vulnerability elsewhere or reduce the amount of resources within the system that can be used for running important application software). Reference [11] doesn't consider one-time attacks (attacks that won't repeat regardless of protection side's actions) and conflicts that will arise with accumulation of protection mechanisms (if a system has a mechanism for installing updates the update itself isn't considered as an additional protection mechanism; in the best-case scenario we will simply have redundancy when one vulnerability is covered by multiple protection mechanisms, nut in the worst-case scenario protection mechanisms will be conflicting for the system's resources).

Reference [12] presents 18 factors that determine the protection level of the system. But it is worth mentioning that the units of measurement for those factors weren't given. Rewards and punishments for compliance/disobeying the security policy for personnel may be expressed in cash (equivalent), while colleagues' and boss' compliance can be expressed in percent from work time (the ratio of time during which a person strictly follows all security rules to the total duration of the working day) or in percent from volume of work (the ratio of actions that are performed in accordance to security policy to the total amount of actions) or even in a boolean variable (when the fact of deviating from security policy *at least once* by *at least one* employee is essential).

III. THE AIM AND OBJECTIVE OF THE STUDY

Discovering effective strategies for protection of the information system from possible attack scenarios is the objective of this study. The monotone decrease of threat level, lack of constraints for the duration of system's functioning and the lack of amplifying effect on the violator were set as the main requirements for protection strategies.

The monotone decrease of threat level means that any protection means and assets that are implemented according to the said strategy must not increase the threat level even temporarily. New attack scenarios may arise if these new scenarios are less threatening than the ones we try to get rid of. This requirement is essential in systems where protection side's budget for security can be considered unlimited (all assets are funded on demand) because with (nearly) unlimited resources the protection side has greater impact on the violator (whether the impact is direct or indirect), which may cause undesirable effect on other information systems. Whenever you pay a ransom for system unlocking/files decryption after your system was compromised with ransomware you are effectively funding further attacks of the violator even if these attacks will target other systems.

The lack of constraints for the duration of system's functioning means that we assume by default the time of functioning and protection of the system in question is unlimited. Systems that are designed for solving a specific task may use strategies presented in this article but temporary solutions should also be considered for such systems.

The requirement for the lack of amplifying effect on the violator isn't equal to the requirement for the monotone decrease of the threat level because the threat level may temporarily increase when violators form a group or due to quantitative/qualitative lack of protection mechanisms in the system.

A. Multifactor violator model

In multifactor model the violator is represented as a set of factors. Violator's experience, available resources, amount of accomplices, time that is available for conducting an attack, level of access to the system(remote access, physical access, indirect access via an accomplice) are examples of factors.

These factors are not independent. For each pair of factors one can either define a link/connection between them or state the obvious lack of such link. We only deal with pairs of factors within the scope of the model because it would quickly become cumbersome otherwise. For n factors there are $\frac{n\cdot(n-1)}{2}$ pairs of links(or stated absence of links). And if we assume that each link has a direction (a factor that affects and the factor that is affected) this number can be doubled since two factors can affect each other. If we wanted to take triplets into account our model would become more complicated and non-linear, because now we have to examine how the factor is affected by the pair of factors. If the resulting effect is equal to the sum of effects, than we don't need to examine triplets (and other larger clusters of interconnected factors). If there are some minor synergy effects we still can use the multifactor model with pairs of factors and ignore triplets. And if the synergy effects are too weighty we still need to take pairs of factors into account.

Factors that have no links with other factors may be temporarily removed from the model. These isolated factors can be examined later when we need to improve the model's level of detail by adding new factors to the model so that the aforementioned factors are no longer isolated.

Representing a violator with the model that utilizes a set of factors allows formalizing the problem of analysis of violator's abilities. Expert's view/idea of violator is subjective. Such representation allows collecting essential information about violator from the expert. If a certain attack doesn't require accomplices then the factor *available accomplices* won't be used in the model and the experts will account the fact that the violator will have to work alone in that particular case even if he can recruit accomplices in other cases.

B. Attack scenarios

If violator doesn't fulfill the requirements for a certain variant of behavior due to his factor values, the following options are possible:

1) violator can perform an action, but chance for success will be either zero or lower than it would be with sufficient factor values (e.g. attempting to hack into system without enough time to brute force the

password, attempting to download a huge file without enough bandwidth etc.)
2) violator can perform an action, but the loss caused by it is lower or zero (e.g. keeping the system down with denial of service attack for a smaller period of time, encrypting fewer files with ransomware)
3) a combination of 1) and 2) (lower chance for success and lower impact in case of success)
4) violator may be unable to perform a given action and forced to choose another option (if alternative options are available).

Condition-factors have fixed values for the entire duration of attack. The values of resource-factors change after each step of attack. Violator' experience (within a single attack scenario the changes are negligible), time to start an attack and violator's hardware are examples of condition-factors. Time that is available for attack and alarm levels (each suspicious action in the system will affect the current value of suspiciousness for violator, therefore violator can only perform a limited amount of actions before the system recognizes and labels him as a violator and blocks his access) are resource-factors.

We will divide all violator's available attack scenario into a few subsets. Each subset has scenarios of a certain type (the same sequence of *types* of actions). For example we can have a subset of attack scenarios that contain *reconnaissance* as one of the steps and a subset of attack scenarios that are conducted without it. Similarly we can highlight a subset of scenarios that contain a step of *replacing traces* after attack.

Each subset of attack scenarios can be represented as a table where columns correspond to steps while rows contain variants of behaviour for each step (the amount of available variants of behaviour may differ for various steps – e.g. there can be 10 variants of *reconnaissance* and only 3 variants of *replacing traces*).

In that case a scenario is a sequence of cells that contains *at least one* cell from each column. If violator picked an action that wasn't effective at current step (either due to improper factor values or for other reasons) violator may try to perform that action again or pick another action from this column. During this process the resource-factors are spent. If violator has not enough resources to continue the attack, the scenario is aborted, and the impact will depend on successful actions from previous steps (if an action is harmful by itself out of context of scenario).

Spending resource-factors will reduce a set of available actions at each step. Successfully performing an action ,ay replenish the values of resource-factors (e.g. resetting the alarm level via shutting down security mechanisms, gaining extra time for attack via diversion, that will draw security officers' attention from the attack in progress).

Reduction of the amount of available scenarios improves our chances to predict an attack in the future (as we can pay more attention to attacks that are still possible). Even if a certain attack scenario can't be completely excluded from the model, it will be possible to prepare for that attack carefully.

Scenarios that by default have more than one cell from one or more columns selected are allowed. Within the scope of the scenario multifactor model factors' values are the only limitation. Such scenarios will be typical for violators who attack our system for the first time (when they don't know what vulnerabilities can be exploited and count that at least one of known vulnerabilities will be present in the system) and have sufficient resource-factors to conduct an attack. Successfully completing at least one action for the given step is required before moving on to the next step in the scenario. To counter such attacks the protection side should figure out how the system that is being protected will react to successful completion of multiple actions within a single step. As it has already been mentioned before, successfully completing a step of scenario may partially or completely restore resources that were spent for that step.

Restoring/replenishing the resources with *an excess* within a single step is possible but it isn't considered in this article. If a certain step allows violator replenishing the resources with an excess proceeding the attack becomes pointless because it is profitable for violator to simply repeatedly reach that (non-final) step and collect the profit. Since the possibility of success for any violator's action can't exceed 100% and with trained personnel and basic protection mechanisms implemented it will be lower than 100%, each step of attack poses a risk of losing resources for the violator (even if he isn't spending money and stays outside the reach of protection side's and law enforcement's influence, it is impossible to avoid spending time that could be used to attack other systems). In the limiting case, when the very first step of attack allows replenishing resources with excess, proceeding with attack becomes pointless for *economically motivated* violator.

The protection side should understand how exactly the replenishment of resources works in case of successful completion of multiple malicious actions within one step. The following variants are possible:
1) independent replenishment
2) (automatic) choice of maximum replenishment
3) choice of replenishment
4) replenishment by the first successful action of the step
5) replenishment by the last successful action of the step
6) random choice of replenishment
7) non-contradictory/uncontroversial replenishment

In case of independent replenishment of resource-factors violator's profit from an attack at any given step is calculated as a sum of profits from all successful actions of that step. Independent replenishment is typical for independent scenarios when compromising successfully a fragment of the system won't affect other parts of that system and won't affect the progress of other scenarios. Independent replenishment extremely dangerous and undesirable for protection side because it prompts violator *to attack more*.

There is a vital difference between *a single violator* who conducts n attacks and a set of n violators, that are not linked in any way between each other, performing those same n attacks because in case of a set of violators a certain subset will fail in their attacks (inevitably wasting resources) and lose an ability to conduct new attacks on our system (or at least the frequency of attacks from those

violators will decrease), while violators who succeed with one attack may still fail with the next one (whether they use the same scenario or not).

With only one violator who conducts n attacks, the threat won't decrease. It is worth mentioning that violators who are capable to conduct multiple attacks at once are, by definition, more dangerous, since they (inevitable) have more resources, but this isn't the only problem. If a system has no protection mechanisms or if those mechanisms can't ensure sufficient level of protection, an expected value of profit from an attack for the violator will be a positive number. The violator doesn't need to succeed in more than 50% of attacks; it will be enough to have an income from successful attacks exceeding the costs for/of all attacks (successful and failed). In case of a set of violators that aren't linked between each other, a certain subset will lose an ability to attack our system (either at all or in near future) in case of failure, while violators who succeed will receive profit from attacks. But this profit will be fragmented (divided between the whole group), and for each individual violator there may be not enough resources to move on to more threatening, complicated, costly attacks. And the threat from a single violator who conducted the same n attacks will increase!

Automatic choice of maximum replenishment means that only one action from this step of scenario determines the replenishment of resources for the violator. Success of the most destructive/harmful actions overlaps success from all other actions. If there is only one resource-factor (or if only one resource-factor can be replenished within each step), estimating the losses for protection side will be easy. We just use a function that returns the maximum value from the set. For all other cases we have to set the ratio of values of resource-factors.

Although replenishment of maximum amount of resources is undesirable, the protection side can use the situation for its own benefit. According to the model, success in one scenario can make another scenario insignificant. For example, compromising application software is pointless for a violator if he has already compromised an operating system on the host where that application software runs. This allows protection side to focus on protection of the key segments of the system, since the implemented security mechanisms for other segments of the system can be bypassed when these key segments are compromised.

The choice of replenishment allows violator choosing the influence on the system by himself. The violator is forced to choose only one option in this case. The violator may choose a less destructive attack (or attack step), if these actions will replenish a resource-factor that is currently important for the violator, even if there are other options where the total cost of replenished resource-factors is higher. This can be observed when violator needs to sustain an attack in progress (to prevent its interruption by protection side or by any external forces) or during parallel attacks on multiple systems (when the resources replenished in attack on our system are used to launch/sustain an attack on a different system). It is worth mentioning that we as the protection side can only observe interaction of the violator with our system.

Replenishment by the first successful action of the step is typical when violator needs to conduct an attack *fast*. The violator won't wait for successful completion of other actions even if the profit from them may be higher and the chance of failure is insignificant. There may be various reasons for such haste: an attack may be available only for a short period of time or it may be conducted to draw attention away from another attack in progress or to divert system's maintenance personnel's from restoring the system after previous attack.

Utilizing honeypot as a protection mechanism (only in countries where they are allowed), or virtual machines and sandboxes allows protection side diverting violator from more destructive actions.

Replenishment by the last successful action of the step is typical for interactive attacks when the protection side and violator are continuously involved, with protection side actively trying to interrupt any attack scenario. The violator tries to find the *most convenient* vulnerability in the system. The profit from completing the step isn't determinative but the action for that step must be successful. What may possibly prompt a violator to interrupt an attack scenario that was successfully started? It may be low reward in case of successful completion of that attack step. The violator may be not aware about type and quantity of resource-factors that will be replenished in case of success. If the result of conducting a certain attack step is unsatisfactory for violator and if he still has resources, he may try another scenario but counteraction from the protection side means he can't return to the previous one.

Random choice of replenishment is possible in 2 cases. In first case the violator may be unaware about of replenished factor-resources for successful steps of attack scenario but there is a way to tell which actions were successful. Additionally the violator may choose blindly with which attack he wants to proceed (so it is a bit similar to choice of replenishment case). In second case the violator may know exactly the amount of factor-resources that will be replenished for each step if only one attack is in progress. Attempting to conduct multiple attacks may cause them to either enhance or weaken one another. A system that is being affected by one piece of malware may become invincible to other attacks. If the system is currently inaccessible due to denial-of-service attack, compromising its integrity and/or confidentiality will be impossible, because nobody (including violators) cant access such system. Similarly, stealing data (violating confidentiality) that was previously damaged/corrupted (violating integrity) will be pointless. Violating availability may also become pointless because then nobody will find out that the information was damaged (e.g. if the attack goal was to harm reputation of the organization). Even if attacks are targeted at a single property of information (confidentiality, integrity, availability) or even at a single piece of information (an account of a certain user or a specific file), malware that is used for attack may compete/conflict with other malware for resources (e.g. a few different encryption

viruses or a few crypto currency miners will conflict for CPU time).

Non-contradictory/uncontroversial replenishment means that the operation of *logical OR* will be used to calculate the amount of replenished resources-factors from the successful step of the attack. This can be demonstrated with the following example: let's assume that our system has 100 files that are equally important. Compromising any single file will replenish n units of a certain resource-factor. Two malicious actions were done during that particular attack step. The first action affected the first 70 files and a second one affected files through 50 to 90. In that case $90 \cdot n$ units or resource-factor will be replenished in total.

The protection side's goal is rapid reduction of available attack scenarios and reduction of impact from remaining scenarios.

There are three approaches to achieve the aforementioned goal:

1) increasing the values of condition-factors for violator (e.g. creating/utilizing defense mechanisms that would make attack impossible without accomplice or a certain amount of accomplices, hiring more security personal to patrol the area to minimize the time that can be spent by violator near the terminal, using longer passwords etc.)
2) increasing the rate of resource-factors spending (forcing the violator to spend more time for preparation to attack at the cost of less time for conducting attack itself; implementing steganography, hiding valuable information in a large heap of information garbage, damaging files intentionally so that violator will waste time on restoring them, filling bandwidth with useless traffic, coding information with high redundancy);
3) decreasing the amount of resource-factors that can be replenished by violator after successful completion of certain steps of attack scenario (post-attack restoration mechanisms, system monitoring, diversion recognition).

C. Choice of protection strategy

The three approaches of minimizing loss that were presented in the previous chapter have their own advantages and disadvantages. *Increasing the values of condition-factors for violator* allows the protection side to sift out weak, ill-organized violators and small groups of such violators but any violator or group of violators with suitable values of condition-factors can start and conduct a complete attack. Increasing the values of condition-factors also means that some attacks will be delayed due to the time required for preparation. If *increasing the values of condition-factors* is the only method of protection, then the protection side must be certain about correctness of restriction enforced by those factors' values (e.g. an attack that requires 2 accomplices should be impossible with less than 2 accomplices under any circumstances). Availability of an alternative set of condition-factors that is less restrictive is unacceptable.

This approach is reasonable when the fact of attack's occurrence is essential regardless of its impact. If the system doesn't allow any deviation from regular mode of operation (e.g. systems that can't be restored after attack), decreasing the amount of violators who can conduct an attack *here* and *now* is the best strategy.

Using the opposite approach (by decreasing purposefully the values of condition-factors) expecting a conflict of violators when a few attacks occurring at the same time obstruct each other is too risky and situational.

We should also consider a special case when attack consists of a few steps with contradictory values of condition-factors. This can be demonstrated with the following example:

Let's assume that our system is a C class network (it has $2^8 - 2 = 254$ adresses for hosts, one address is reserved for broadcast and one address is the network address itself). Our router is configured to drop all traffic from Internet. But this router "knows" to which ports these 254 hosts are connected (any message with a forged header of the packet, that resembles a message from within the local network is also dropped if it was received from the wrong port). Violator's goal is to ruin the integrity of information that is stored on these 254 hosts. Violator can't be present near each of 254 hosts at the same time and this attack can't be performed with remote access (so that an attack must be launched from within the network from one of this network's hosts). But there is a problem with this plan since there are already 254 hosts in this network. In order to connect to the network a violator has to:

1) disconnect one of those 254 hosts
2) connect to the network temporarily using the IP-address of the aforementioned host
3) conduct an attack on 254 hosts

The disconnection of the host during the 1st phase of this attack may be discovered by implemented protection mechanisms. If IP-addresses are static then during 2nd phase violator needs to find out the disconnected host (if it has been unknown by this moment; this can be done with ipconfig/ifconfig command for Windows/Unix systems). Violator can conduct an attack using a broadcast but then his own host will be attacked. If the first two phases were completed successfully we can still observe an interesting problem. The host that was (temporarily) disconnected from the network won't be attacked! If all 254 hosts must be attacked simultaneously it will become impossible due to a single disconnected host.

In systems with two (or more) levels of protection a violator can't know the values of condition-factors for successful completion of the next protection level. Reconnaissance can only give insight about condition-factors for the first level of protection mechanisms. In preparation to attack such system a violator has to "prepare for everything" which will delay an attack.

The strategy of *increasing the rate of resource-factors spending* should be used when the level of condition-factors is too low and can't be raised via implementing new protection mechanisms/improving the existing protection mechanisms. In this situation protection side deals with a case where attack probability (or the probability of the first step of such attack) is close to 100%. We are interested in reducing the available time for attacks that affect information's confidentiality and integrity (the less time violator has to interact with sensitive information the less

harm can be done; less information can be stolen with limited bandwidth, less information can be corrupted with malware/ransomware). Using longer and more complicated passwords belongs to this strategy (violator spends less time logged in), but it isn't the most effective protection mechanism. Aside from the most obvious flaws (difficulties of memorizing such passwords, input errors when typing the password, writing down the passwords on paper and storing it on your desk) we can't guarantee that any resources will be spent to brute force the password.

Choosing the strategy of *increasing the rate of* violator's *resource-factors spending* protection side can't get away with just creating conditions for spending more resources. It is necessary to force violator to complete a costly attack. Protecting *all* assets within the system is crucial for this strategy (not just information that is processed in the system but also software and hardware) so that violator can't choose more approachable target for his attack.

Economically-motivated violators base their attack target on cost of attack to potential gain ratio. Other non-economical motives (e.g. revenge, self-affirmation) can affect this ratio but the fact of motivation for attack is still important Protection side who decides to use the strategy of *increasing the rate of resource-factors spending* should sustain violator's motivation/interest so that he won't stop attacking. This can be achieved with following methods:

1) increasing the cost of assets (the most obvious flaw here is the bigger loss if violator succeeds);
2) misinforming violator about assets value;
3) reducing protection levels (milder restrictions may prompt violator to conduct an attack even if takes a few hours or even days);
4) misinforming violator about protection level (this works for overstating and understating the actual quality of protection mechanisms; unlike previous item from this list, we don't need to take actual risk of reducing protection levels; violator can be misinformed about fake deadline of removing some vulnerability or deadline of information remaining valuable, provoking him to attack the system in a more convenient/suitable time for us);
5) handing in misinformation disguised as system assets (violator may believe that attack was successful and it is reasonable to repeat it);
6) misinforming violator about responsibility for attack (our abilities to track and pinpoint violator's location and identity, ability to bring him to accountability, our abilities to launch a counter-attack, etc.)

Strategy of *decreasing the amount of resource-factors that can be replenished after successful completion of certain steps of attack scenario* is used to reduce the consequences of successful attacks and to reduce the probability of such attacks in future. This includes dealing with ransomware (blocking your system or encrypting your files with further ransom for the decryption key) and seizing the resources of the system (botnets and crypto-currency miners). Refusal to pay ransom for unblocking the system and shutting down compromised nodes of the system will reduce violator's income from conducted attacks. Of course those actions can be harmful for the system in question. Sometimes it is cheaper/faster to pay ransom and purchase the key to unblock your system from violator than to restore the system form back up copy. Afore mentioned examples show us that using this strategy at final steps of attack can be undesirable. This is why one should implement protection mechanisms for *intermediate* attack steps. Implementing quotas for memory usage and processing power will limit effectiveness of crypto-currency miners and similar software, because in case of system's compromise only a fixed fraction of resources will be lost that may still allow a system to operate in regular mode.

This strategy can also be used to choose protection mechanisms from attacks on *availability*. Denial-of-service attacks are different from attacks on confidentiality and integrity because the time of conducting these attacks must be maximal (violator benefits more from keeping our system un-operational for longer periods of time). If implemented protection mechanisms support at least minimal functionality of the system or allow the system to operate offline for some time without vital losses, such attacks won't be effective.

System monitoring mechanisms and attack detection/consequences of attack detection will also reduce the amount of replenished by violator resource-factors. Detecting a preparatory phase of attack (system scanning, data collection via social engineering, sending insiders to our organization) allows us preparing for further steps of this attack (configuring IDS/IPS to track certain types of events in the system, adding information of violator's interest to honeypots, transferring and rearranging personnel, installing cameras for hidden observation to track personnel, changing passwords and/or requirements for passwords' length and complexity, if they are considered too weak, revoking digital signature certificates, changing system operating and maintenance schedule, etc.).

D. Alternative and irreplaceable resource-factors

The main problem for the second and third strategies is the violator's ability to redistribute available resource-factors. The lack of knowledge and experience can be compensated to a certain extent with the abundance of free time, recruiting extra accomplices, and even additional hardware. Protection side should account for the "exchange rate" of resource-factors that can change over time and differ for various types of violators. To simplify the model we will need to use the following restrictions:

1) The exchange of resource-factors is always direct (there are no transitional links). If resources A,B and C are available for violator and he needs more C, then exchanges $A \to C$ and $B \to C$ are allowed but $A \to B \to C$ is not;

2) The effectiveness of resource exchange can't exceed 100% so that violator can't generate more resources "from thin air" just by exchanging them in a specific order (the only exception is when the new resource is irreplaceable and can't be converted further);

3) The resources are discrete values (not necessarily integer numbers but there is indivisible unit of resource that determined the minimal amount of resource that must be converted within one transaction);

4) During conversion the result is always rounded down (this is why it is hard to achieve even 100% conversion effectiveness; with 3:2 exchange ratio the violator will receive 2 units of resource B for 3 units of resource A, but only one unit of resource B for two units of resource A instead of 1,5 units of B if B can be represented only with integer numbers).

Though it is not necessary, preservation of relative values of resources received from exchange is desirable.

TABLE 1: relative Values of Resources C and D

	A	B
C	20:1	4:1
D	15:1	3:1

a – relative value is preserved

	A	B
C	20:1	4:1
D	15:1	1:1

b – relative value isn't preserved

Tables 1.a and 1.b depict the exchange rate for A and B resources for C and D respectively. In case of 1.a the value of resource D is set as 75% of resource C and it doesn't matter which resource (A or B) will be used for exchange. In case 1.b the value of resource D varies between 25% and 75% of resource C value. In the first case we can introduce some universal unit of exchange (e.g. the cheapest resource or the fraction of it) to determine the overall quantity of resources. The protection side may be unaware about exact distribution of violator's resource-factors, but it may be easier to evaluate the overall quantity of resources. After implementation of protection mechanisms the overall quantity of resources for successful attack should be approximately equal for all attacks of the same type. If protection side is aware of attack scenario with high chances for success but with significantly lower requirements for resource-factors this indicates the presence of vulnerability in the system that must be removed urgently.

IV. CONCLUSION

Violator's choice of attack depends on available resources and possibility of resource redistribution. The values of *condition-factors* don't change during attack and determine strict restrictions for violator's actions. The protection side is interested in *condition-factors* because they determine the compatibility of attacks. The system that is currently under Denial-of-service attack may be unavailable for attacks that violate its *integrity* or *confidentiality*. Condition-factors for system that is in a state of recovery after crash/malfunction, update or creating a backup, will be different from condition-factors of the system that operates in regular mode.

The strategy of *increasing the rate of resource-factors spending* allows interrupting an attack before its completion, while the strategy of *decreasing the amount of resource-factors that can be replenished after successful completion of certain steps of attack* will restrict the overall quantity/frequency of attacks.

Further research should be focused on discovery of irreplaceable resource-factors and methods of altering the system in a way that will either force violator to rapidly spend these irreplaceable resource-factors or prevent their recovery.

ACKNOWLEDGMENT

I would like to thank my family and friends for their support. I also want to thank prof. O. Arkhypov for suggesting this topic for my research.

REFERENCES

[1] Yu. M. Polekhina, D. S. Tymofeiev (2010). Model Porushnyka. Meta ta Pryntsypy Rozrobky. *Sovremennyye informatsionnyye tekhnologii.* [Online]. Available: http://www.rusnauka.com/11_EISN_2010/Informatica/63866.doc.htm

[2] D. S. Biriukov, V. A. Zaslavskyi, V. V. Yevhiienko and O. V. Franchuk, "Modeliuvannia ta Otsinka Stsenariiv Zahroz dlia Obiektiv krytychnoi Infrastruktury," *NAUKOVI ZAPYSKY,* vol. 99, pp. 97-101, 2009

[3] V. L. Buriachok, "Model Formuvannia Dereva Atak dlia oderzhannia Informatsii v informatsiino-telekomunikatsiinykh Systemakh i Merezhakh pry vyluchenomu Dostupi." *Informatyka ta matematychni metody v modeliuvanni,* vol. 3, №2, pp. 123-131, 2013

[4] M.M. Voitko, "Pobudova uzahalnenoi Modeli Zahroz dlia System Internet-bankinhu." *Financialspace,* vol. 3(15), pp. 33-38, 2014

[5] I. V. Kotenko, M. V. Stepashkin (2013). Modeli Deystviy Khakerov-zloumyshlennikov pri Realizatsii raspredelennykh mnogoshagovykh Atak. [Online]. Available: http://masters.donntu.org/2013/fknt/zhadanov/library/kotenko_z.pdf

[6] *Metodika Opredeleniya Ugroz Bezopasnosti Informatsii v Informatsionnykh Sistemakh*, Metodic Document 2015.

[7] M. Bergh, K. Njenga, "Information Security Policy Violation: The Triad of Internal Threat Agent Behaviors," *Proceedings of the 1st International Conference on the Internet, Cyber Security, and Information Systems (ICICIS),* Gaborone, 18-20 May 2016.

[8] A. Loukaka, S. M. Rahman Shawon, "Discovering new Cyber Protection Approaches from a Security Professional Perspective," *International Journal of Computer Networks & Communications (IJCNC),* vol. 9, №4, pp.13-25, July 2017.

[9] S. S. Park, A. B. Ruighaiver, S. B. Maynard and A. Ahmad, "Towards Understanding Deterrence: Information Security Managers' Perspective." *Proceedings of the International Conference on IT Convergence and Security 2011, Lecture Notes in Electrical Engineering 120,* December 2012.

[10] E. T. Jensen, "Cyber Deterrence." *Emory international law review,* vol. 26, pp. 774-824, May 29, 2012. [Online]. Available: https://ssrn.com/abstract=2070438

[11] H. Mouratidis, P. Giorgini and G. Manson, "Using Security Attack Scenarios to analyse Security during information Systems Design," *Proceedings International Conference on Enterprise Information Systems,* pp. 10-17, Porto, Portugal, 2004

[12] S. Abraham, "Information Security Behavior: Factors and Research Directions." *AMCIS 2011 Proceedings - All Submissions, 462,* [Online]. Available: http://aisel.aisnet.org/amcis2011_submissions/462

Oleksandr Kireienko.
Was born in Ukraine, 1993, 7th March.
Finished National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" in 2016 and obtained master's degree in "information and communications systems security" with professional qualification "information security professional"
He has worked for three years in National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" as an Assistant and a 4th year Ph.D student at NTUU KPI. He has published the following articles:
[1] O. Kireienko "Violator Model for Information and Communication Systems" *Legal, Regulatory and Metrological Support of Information security system in Ukraine,* vol. 34, №2, pp. 69-77, 2018.
[2] O. Kireienko "Multifactor Violator Model for Information Security" *Legal, Regulatory and Metrological Support of Information security system in Ukraine,* vol. 35, №1, pp. 61-69, 2018.
[3] O. Kireienko "Information Security violator model with general and specialized information," *Ukrainian Scientific Journal of Information Security,* vol. 25, №1, pp.24-29, 2019.

[4] O. Kireienko "Multifactor Violator Model with Information Gathering," *Legal, Regulatory and Metrological Support of Information security system in Ukraine,* vol. 36, №2, pp. 7-14, 2018.
Research interest include information security and game theory.