Anticipating Disasters through a Security Twin

Fabrizio Baiardi [1*], Salvatore Ruggieri [1†] and Vincenzo Sammartino [1†]

[1*]Dip. di Informatica , Universita di Pisa,
Largo B.Pontecorvo, Pisa, Italy.

*Corresponding author(s). E-mail(s): f.baiardi@unipi.it;
Contributing authors: s.ruggieri@unipi.it;
vincenzo.sammartino@phd.unipi.it;
†These authors contributed equally to this work.

Abstract

A security twin is a specialization of a digital twin focused on the security and robustness properties of an ICT/OT infrastructure. By running adversary emulations that use a security twin, we can discover how a hybrid threat can exploit vulnerabilities to build intrusions to penetrate and control critical infrastructures. According to the target infrastructure, the control by a threat may result in a disastrous impact on the whole society. Information on possible intrusions is fundamental to anticipate potential disasters and to increase the resilience of a (digital) society. Due to the highly dynamic evolutions of both infrastructures and hybrid threats, we advocate that only adversary emulation using a security twin can convey the information to forecast intrusions and prevent social and economic disasters. We also discuss how to execute adversary emulations and how a security twin is built and updated.

Keywords: security twin, adversary emulation, data shift, elementary attack, fault

1 Introduction

The advent of ICT/OT networks has transformed how we access information, control supply and production chains, and make decisions, with almost instantaneous data dissemination across networks, whether it is news distributed online or confidential information distributed in an intranet. While this interconnection has led to significant economic growth and improved social connectivity, it also brings the risk of

malicious misuse, causing harm to vulnerable or unsuspecting users [1–4]. As critical information infrastructures, or simply infrastructures, become more interconnected and efficient, they also become more susceptible to exploitation, resulting in greater harm to larger populations and user groups than before. At the heart of this issue lie hybrid threats that attempt to disrupt or distort the infrastructure components to damage the public or specific stakeholders within government, corporate, or other organizational contexts. Interdependency among infrastructures may further increase the overall damage [5]. Both state and non-state actors can instigate hybrid threats to create social, economic, or organizational discord within targeted groups. The threats can exploit several distinct vulnerabilities ranging from those in software and hardware components to those in users and administrators of an information infrastructure. The effectiveness of their cyber attacks increases if the threats also implement physical or phishing attacks against the infrastructure. The difficulty of designing solutions to these problems increases because of the complexity of infrastructure and the lack of models and methodologies to discover how the current vulnerabilities enable hybrid threats to reach their goals.

Both an accurate assessment of the risk hybrid threats pose and the prediction of their intrusions are fundamental for disaster preparedness and management. Unluckily, in most cases, preparedness and management are not up to the challenges these threats pose because of the complexity of discovering intrusions and estimating the probabilities to manage the resulting risk. Furthermore, the risk scenario is highly dynamic and frequent data shifts occur that reduce the usefulness of historical data to estimate these probabilities. Due to the lack of proper formal models to quantify these probabilities, we propose substituting historical data with synthetic data generated through adversary emulations of the incursions by hybrid threat actors. Drawing inspiration from digital twin technology in model-based engineering [6], these emulations utilize executable models, or *digital twins*, of both the target infrastructure and the actors. Unlike physical object twins, the twin of the target infrastructure is more abstract, focusing specifically on security and safety concerns rather than providing a detailed virtual replica of the infrastructure. Hence, we denote it as a *security twin*.

Sect. 2 of this paper briefly describes the information of interest on intrusions by hybrid threat. Then Sect. 3 discusses how to describe intrusions and Sect.4 reviews the issues to predict these intrusions. The following sections describe the solution we propose. Sect. 5 describes how to discover intrusions using twins, Sect. 6 focuses on the building of twins and Sect. 7 on how to use the output of the adversary emulations to discover possible intrusions.

2 Resilience to Hybrid Threats

We briefly describe how hybrid threats operate and then how we can increase the resilience to these threats [7, 8].

2.1 Tactics, Techniques and Procedure of Hybrid Threats

The intrusions of hybrid threats typically involve a combination of cyber operations and non-traditional methods, such as physical attacks or social engineering techniques. Hybrid threats frequently implement intrusions and cyber espionage to control infrastructures, steal sensitive information, or sabotage systems. Their techniques can include Distributed Denial of Service (DDoS) attacks, malware deployment, phishing campaigns, and data breaches. Hybrid threats engage in psychological operations to influence the perceptions, behaviour, and decision-making processes of target populations. This may involve psychological manipulation, intimidation tactics, or fostering fear and uncertainty among adversaries. Intrusions and psychological operations are strongly related because intrusions may return information that is then manipulated and spread through psychological operations. Simultaneously, psychological operations aim to influence the users and the administrators of an information infrastructure and increase the success probability of phishing and spear phishing attacks.

Hybrid actors exploit ambiguity and maintain plausible deniability by conducting operations through non-attributable means, using proxies, or employing tactics that blur the line between conventional and unconventional warfare. This makes it challenging for targeted entities to accurately attribute attacks and respond effectively. In cyber intrusions, the proxy is an attack infrastructure, i.e. an overlay network with nodes previously attacked in a stealth way. This network can be used as a command and control infrastructure to coordinate malware deployed in a target infrastructure or to execute bots that spread malicious information in social networks [9].

Overall, the TTPs of hybrid threats are adaptive and multifaced. They include a range of conventional and unconventional methods by exploiting vulnerabilities in target infrastructures to achieve strategic goals in society.

2.2 Improving Resilience

Proper countermeasures should be deployed to prevent the potential disruptions of hybrid threats to the infrastructures of national and local governments and companies. A precondition to select countermeasures includes the following objectives: 1) identifying and understanding the sources of disruptions and possible intrusions; 2) discerning the vulnerabilities within these infrastructures the intrusions exploit and their success probabilities; and 3) assessing how successful intrusion produce cascade failures to other infrastructures, production plans, and social harmony and cohesion [10].

This paper addresses the last two objectives. To effectively tackle them, alternative strategies may be adopted, but they all share the same starting point: discovering how a threat actor builds intrusions to subvert an infrastructure, their success probabilities, and the vulnerabilities that enable these intrusions. This allows us to understand how the actor can propagate across the interconnected infrastructures and produce impacts potentially affecting the whole population.

3 Executing and Describing Intrusions

First of all, we introduce some terms used in the following.

We focus on intelligent actors that aim to reach a predefined goal. A goal is a set of privileges, i.e. access rights on resources of an infrastructure. We assume each actor only has one goal and that actors do not cooperate in an intrusion.

An intrusion is the sequence of actions a threat actor, or simply actor, executes to reach its goal and control some physical or logical resources [11–13]. Some actions exploit the weaknesses of the target infrastructure. Other ones, typical of hybrid threats, are based on social engineering and target users modelled as further infrastructure components. An intrusion is successful if, after its last action, the actor reaches its goal – for instance, it can lock, encrypt, delete, or steal the resources it controls. We need to know all the intrusions actors can implement to assess the overall risk.

Alternative actions of an actor may be classified in terms of the Tactics, Techniques, and Procedures (TTPs) [14].

We can describe each action through a tuple with five elements:

<IPs, IPd, attacktec, rights, information>

where:

1. *IPs* is the IP address of the node where the action is executed,
2. *IPd* is the address of the target of the action,
3. *attacktec* is a technique or a subtechnique in the Att&ck matrix [14]. It also includes information about the enabling vulnerabilities if any,
4. *rights* is the set of access rights the actor acquires because of the action,
5. *information* is the information the actor acquires because of the action

As an example, *IPs* may be the address of the node scanning the node *IPd* and *attacktec* is the active scanning technique. The *information* field includes the vulnerabilities of *IPd* while the *rights* one is empty. If the action is a successful attack, this field includes the access rights the attack returns.

Some actions, such as an attack or a vulnerability scanning, can fail and do not return the expected output. If an attack fails, both the information and the rights fields in the corresponding tuple are empty.

In an intrusion, the actor strategy chooses the action to execute at a given step. Alternative strategies exist but all satisfy the same constraint because any actor can execute an action only if it owns the proper access privileges and information. This results in a *consistency condition* on intrusions. According to this condition, a sequence of actions, or a sequence of tuples, S is an intrusion of an actor Ag if :

1. the first action *act* in S belongs to the *attack surface* of Ag, i.e. the attacks enabled by the legal access rights of Ag,
2. the union of the legal access rights of Ag and all the rights returned by the action in S before an action A includes the precondition of A,
3. any action in S always fails if its target is not affected by the enabling vulnerabilities. The two last elements of the corresponding tuple are empty.

Tuples of S that model distinct failures of the same action share the first three elements and their last ones are empty. An intrusion repeats failed actions only.

An intrusion S is successful if at the end of S, Ag owns the access rights in its goal. Otherwise, S fails. An action A is *useless* if the actor does not need the privileges or

the information A grants to reach its goal. This happens if the access rights A grants do not belong to the precondition of any useful action in S. Ag executes a useless action because it has not collected enough information on the target.

We map an intrusion S into a *plan* by removing from S any tuple describing a useless action or a failed attack. Hence, plans define intrusion that cannot be reduced. Plans are important because we can stop an intrusion only by preventing the execution of actions or attacks in the corresponding plan. Distinct plans can reach the same goal and the number of these plans shows the degree of freedom of the actor in building intrusions. As a consequence, this number influences the overall risk. The time to execute all the actions in the shortest plan is the shortest time to reach the goal.

4 Forecasting Intrusion and Data Shifts

This section argues that we cannot discover all the intrusions against an information infrastructure or compute their success probabilities using data collected on past intrusions against the same or similar infrastructures because of data shifts, that is because the underlying probability distributions change too quickly.

To prove our claim, assume we can discover intrusions and evaluate accurately the success probability of intrusions by Ag to reach a *goal*. All this information becomes obsolete as soon as any of the following occurs:

- Ag changes its behaviour and executes new techniques in an intrusion,
- a new vulnerability is disclosed in an infrastructure component,
- some nodes of the target run a new application,
- a new logical or physical connection is created between some infrastructure nodes,
- some nodes are added to or removed from the infrastructure.

The reason for obsolescence is that these events may result in new intrusions and/or change the intrusion success probability. In other words, these events may produce a data shift in the distribution of the data to compute the probabilities of interest [15, 16]. In more detail, new behaviours of Ag change either its TTPs or the sequence of actions in its intrusion. As an example, Ag may increase the time it devotes to collecting information before selecting the vulnerability to exploit. This implies Ag collects more information to solve its choices and avoid useless attacks. This reduces the number of attacks in its intrusions and/or the number of failures. We can update Ag success probability after several intrusions, but we cannot ask Ag to repeat a successful intrusion just to compute more accurate probability values.

Further data shift is due to the discovery of a new vulnerability, the deployment of a new application, or a new connection. All these events may enable new attacks that, in turn, result in new intrusions. The corresponding impacts cannot be predicted using data on previous intrusions because multiple vulnerabilities increase in a nonlinear way the number of intrusions.

We conclude that any framework to evaluate accurately the number of intrusions or their success probability should handle frequent data shifts that affect the risk scenario. These shifts are very frequent and prevent the collection of a proper amount of actual data on intrusions. This lack of information may result in a black swan or a perfect

storm disaster [17, 18] that is in intrusions that are unknown in advance or known but unexpected because their probability cannot be estimated with the required accuracy. In most cases, even the overall social impacts of these intrusions are unknown.

5 Using a Security Twin To Discover Intrusions

As discussed, data shifts prevent an accurate forecasting of future intrusions. The lack of actual data could be recovered by using a formal model of the target infrastructure that could return the information of interest about intrusions. However, any real-world infrastructure is too complex for such a formal approach.

We investigate the behaviour of hybrid threats using synthetic data produced by merging adversary emulation and twin. Adversary emulation reproduces the behaviour of threat actors but against the security twin rather than against the real system [12, 19]. This is inspired by the digital twin technology, a branch of model-based engineering used in other fields [20–23]. Multiple twin-based adversary emulations produce a large amount of data on alternative intrusions. Information can be distilled from this data to avoid or mitigate the impact of hybrid threats.

5.1 The Security Twin

According to [24] a digital twin is

> "a virtual model for a physical entity in the digital form to simulate entity behaviours, monitor the ongoing status, recognize internal and external complexities, detect abnormal patterns, reflect system performance, and predict future trends".

This definition stresses that a twin is a model rather than a replica, that abstracts the entity it models and represents some of its attributes only.

A security twin is an enriched inventory of hardware and software infrastructure modules to model accurately the behaviours of, respectively, an actor and the infrastructure in an intrusion. For this purpose, the security twin describes:

a) the infrastructure nodes and their connections,
b) the infrastructure modules and the operations each defines;
c) the mapping of modules onto nodes and the corresponding configurations,
d) the accounts on each node,
e) any vulnerability of each module or node, the attacks it enables, and the properties of these attacks,
f) routing and filtering rules
g) the logical connections among modules the previous rules determine,
h) intrusion sensors, the subnet or the endpoint each sensor monitors, and the probability it detects an intrusion,
i) hierarchical relations, i.e. the one between a hypervisor and the virtual machines it runs
j) information flows among the modules, i.e. one from a web server to a database.

Information on the configuration of the various modules is critical because each configuration may result in distinct vulnerabilities. The operations of a module determine the access rights the actor can acquire in an escalation, a distinct right exists for each operation. Attack properties include the success probability, the execution time, the noise it generates, and pre and post-conditions. The pre-condition and the post-condition include respectively, the rights an actor needs to execute the attack and those a successful attack grants. Pre and post-conditions jointly determine the sequences of actions the actor can execute and they are deduced from information in several vulnerability databases [25, 26].

Information in the twin supports the mapping of the current privileges of an actor into the actions it can execute. These actions are also constrained by the information on the system the actor has acquired. Hierarchical relations and information flows determine dependencies among modules. As an example, some access rights on a hypervisor result in the control of the virtual machines it manages. In an information flow, access rights on the source of a flow enable the manipulation of the values transmitted to the receiving module(s).

A security twin also describes user classes and pairs users in each class with attributes such as their access rights or the probability one of these users is the victim of an impersonation attack. The attack may be implemented by stealing authentication information or by social engineering techniques such as phishing or spear phishing.

Distinct security twins of the same infrastructure differ in the number of details on the modules, the granularity of operations and hence of access rights. Further details concern the behaviour of some components. For example, a twin that neglects intrusion sensors returns worst-case results because it cannot model intrusion detection. A more accurate twin can model the detection and the corresponding failure of an intrusion.

However, any security twin should minimize the overhead of adversary emulations. Hence, these twins never describe in full detail the system, its inputs, and its computations.

5.2 The Actor Twin

This twin describes the actions of an actor, its strategy, the initial access rights and information, and the goal. The attack surface includes any attack enabled by the initial access rights the actor owns before starting an intrusion. The alternative actions of an actor describe how it collects information and the attacks it can execute [14]. Any action is paired with the privileges and the information on the infrastructure it requires and those it returns, if any.

Even this twin is an abstract description of an actor because we are focused on computing the intrusions it can implement and their success probability. Hence, we focus on the output and the success probability of action rather than on execution details. From our perspective, the access rights to execute an attack and its success probability are more interesting than the actions to execute the attack.

The strategy of an actor [11] maps the current status of the actor, its goal, and the target system into the action to execute. If this action is an attack, the strategy also returns a target module and the vulnerability to exploit. The status includes the access rights and the information the actor has collected. As an example, a strategy prefers

information collection to exploitation and it selects an attack only when it cannot collect further information. Instead, other strategies execute an attack as soon as the actor owns the privileges in the precondition of the attack. This may speed up the escalation at the expense of useless attacks. Some strategies include social engineering attacks while others only exploit the vulnerabilities in the infrastructure modules.

The actor status includes not only privileges and information but also a memory that records the last actions of the actor and their success or failure. A small memory implies that the actor forgets after a short time its failures and may repeat an action, or a sequence of actions, even after a large number of failures.

The mapping from states to actions may be expressed as a program, a set of rules or even by a neural network

5.3 Adversary Simulations

In the following, we use simulation rather than emulation because attacks and some actions can be successful or fail according to the features of the enabling vulnerability.

The simulation results from the co-evolution of the security twin and of the actor one starting from the actor attack surface. A co-evolution is a sequence of steps where each one applies the actor strategy, executes the selected action, and determines its success or failure using the information in the security twin. Each step updates the actor status according to the success or the failure of the corresponding action. Lastly, a step considers the reaction of the infrastructure to the actions of the actor. As an example, an intrusion sensor may discover an attack and this results in the failure of the action and maybe of the intrusion. Each step also increases the elapsed time of the one to execute the chosen action.

The simulation of an intrusion may also update the status of the security twin because some attacks could change the infrastructure. The corresponding step updates some twin attributes to model the change. As an example, an actor can update routing or filtering rules and this changes the logical topology in the security twin. Instead, a physical attack to a connection changes both the logical topology and the physical one.

A simulation ends when the actor cannot select an action or it reaches its goal. An intrusion may fail when a predefined number of steps has been executed without reaching the goal, a predefined threshold on the simulated time has been reached or the infrastructure detects an attack of the intrusion.

A simulation is consistent if the corresponding sequence of actions satisfies the consistency condition as defined in Sect. 3.

6 Building and Updating Twins

We describe how twins are built and updated. The frequency of updates in the cyber world implies that both our twins are frequently updated to mirror changes in the infrastructure or the actor and avoid data shifts. Any update should fire the simulations to discover new attack and their success probabilities.

6.1 The Security Twin

The building of the security twin is a reverse engineering operation that extracts the information of interest from actual components. In practice, we build the twin by enriching a system inventory with information on vulnerabilities and attacks. Inventory tools and vulnerability scanners can discover and return all the information of interest. The information to compute the logical and physical topology is imported from routing and firewall rules while those on the vulnerabilities and the attacks each enables are produced by a vulnerability scanning that maps an inventory of the infrastructure components into a database with information about vulnerabilities and weaknesses. This information can be refined by using multiple databases.

The accuracy of the security twin is critical because it determines the one of information on intrusions and it can be assured by verifying that each twin attribute correctly describes an element of the target system and the other way around. The most general assurance strategy exploits the output of monitoring to check that the information on the infrastructure entities, their connections and their interaction is coherent with the one in the twin.

The security twin is updated according to the evolution of the target infrastructure. For this purpose, we run with a predefined frequency the tools to collect information on the infrastructure to discover any change and fire the building of a new twin.

Further updates to the security twin occur anytime new vulnerabilities are discovered in the infrastructure modules. The information on these vulnerabilities and the corresponding attacks is retrieved from databases that are accessed with a frequency chosen by the owner of the target infrastructure according to his/her risk appetite and the impact of successful intrusions.

New simulations are run to discover any further plans the new vulnerabilities or the infrastructure changes enable.

6.2 The Actor Twin

The information to build the actor twin includes the set of its actions, i.e. the TTPs it applies, the strategy to select the action to execute, and how it handles failures. The goal of an actor is related to some features of the target system. As an example, in an industrial control system, ICS, the programmable logical components, PLCs, that monitor and control a production process have a critical role. Hence, the actor goal always includes some privileges on these components. In a Windows OS environment, actors are interested in privileges on Administration Servers. Further information to build this twin concerns the vulnerabilities the actor prefers or its tactics, techniques, and procedures. Usually, this information is an output of threat intelligence.

Anytime the actor changes its TTPs or its strategy, the twin is updated to run new simulations. A standard format of threat intelligence reports simplifies the automatic update of this twin and the firing of the adversary simulations.

6.3 Faults and Physical Attacks

A hybrid threat may also exploit faults of some components or physical attacks to build its intrusion. Another strategy is a supply chain attack against those that produce some infrastructure modules.

A twin-based approach can model the occurrence of faults and physical attacks by updating the security twin. This update can also occur during a simulation if the actor twin can execute the corresponding actions and the actor strategy can select them. The impact of suspected supply chain attacks may be modelled by adding further vulnerabilities to the security twin. We model in the same way suspected vulnerabilities that are not public yet.

This also shows how our twins can support a what-if approach to discover the potential impact of suspected actions by a hybrid threat.

7 Predicting Intrusions

This section describes how to compute probabilities of interest on the intrusions that the adversary simulation returns. First of all, we introduce some terms used in the following. A simulation returns a false positive when it returns an intrusion that does not exist in the target infrastructure. The definition of a false negative is related to a set of independent simulations. Two simulations are independent if the probabilities that determine the sequence of actions in these intrusions are independent, ie if the probability of an action in an intrusion does not change conditional to (i.e. after knowing) the sequence of actions in the other intrusions. A set of independent simulations returns all the distinct intrusions the actor follows in the set. A set of simulations returns a false negative if does not include a plan that exists in the infrastructure.

7.1 Probabilities of Interest

Besides discovering intrusions, it is also important to determine their success probabilities. To this purpose, we apply the Monte Carlo method and run multiple independent simulations, each corresponding to a distinct intrusion. In each simulation, we collect information on the sequence of actions, and using this information, we approximate the probability of an event as the ratio between the number of simulations in which the event occurs and the overall number of simulations. This approach is correct under the assumption that the security twin is accurate (i.e., that all the vulnerabilities present in the twin exist in the target infrastructure) and that the simulation process only returns intrusions satisfying the consistency condition; under these assumptions, one can formally prove that no simulation produces a false positive.

This methodology enables us to approximate the probabilities of events such as:

1. the actor reaching a goal,
2. a module being attacked,
3. the actor executing a sequence of actions,
4. the actor following a plan,
5. an intrusion taking less than a specified time limit.

We can improve the accuracy of the approximation by increasing the number of simulations. It is important to clarify that this improvement pertains solely to the reduction of stochastic uncertainty in the simulation process, namely, the variance of the estimator, rather than to any enhancement in the security twin's ability to capture all (or most) of the target infrastructure's vulnerabilities (including unknown ones, such as zero-day exploits). This distinction is critical because, even with a large number of simulations, the overall accuracy of the probability estimation remains dependent on the completeness and fidelity of the security twin. The absence of any noise affecting the behavior of the target infrastructure allows us to run simulations on a scale that is several orders of magnitude larger than what a red or purple team could feasibly execute. Even if we assume a team can run any number of intrusions, each intrusion produces noise that most OT systems cannot tolerate, and repeated intrusions by the same team are not independent due to the accumulated target information.

The noise issue becomes particularly significant when intrusions are implemented using an attack platform or a breach and simulation tool [27, 28]. Additionally, a larger set of independent simulations increases our confidence in avoiding false negatives. Even if the security twin is accurate, stochastic factors and the combination of successes and failures in individual actions might prevent an intrusion from ever occurring in the simulation. Thus, by running a sufficient number of simulations, the probability of a set of independent simulations returning a false negative decreases. As an example, experiments on infrastructures controlling production lines that require high cyber robustness have involved running between 10^5 and 10^6 simulations to achieve a probability of 0.999999 for detecting any intrusion. The necessary number of simulations is ultimately determined by the complexity of the infrastructure.

7.2 Proactive Management of Intrusions

The synthetic data the simulations produce can be collected to compute probabilities of interest. These probabilities and the knowledge of possible intrusions are the starting point to support decisions about increasing the infrastructure robustness, discovering ongoing intrusions and selecting proper mitigation or resilience strategies. By properly tuning the number of simulations we can also discover intrusions with a low success probability but with huge impact not only on the infrastructure but also on the social organization of a community or a nation.

The collected data can also be analyzed using big data techniques to discover further information on the intrusions. Among others, the data supports the evaluation of robustness, i.e. how long the infrastructure can resist an intrusion [12] or the probability an actor is successful as a function of the time it has available to reach its goal.

8 Conclusion

Critical information infrastructures that include both ICT and OT components are attractive targets for hybrid threats as any reduction in the availability of the services the infrastructures offer heavily affects a social community and results in social and

economic impacts and potential social turmoils. Hence, a proactive discovery of intrusions is fundamental to anticipating and managing their impacts, minimising their overall risk and avoiding social and technological disasters.

The adoption of a technology merging digital twins and adversary emulation can overcome the lack of accuracy in proactive intrusion discovery posed by strategies that forecast future intrusions based on historical data. These strategies neglect the data shift in a dynamic risk scenario due to the rapid evolution of target infrastructures, the new strategies of hybrid threats, and the continuous discovery of vulnerabilities in infrastructure components.

Acknowledgment. Work partially supported by the European Community H2020-EU.2.1.1 programme under the G.A. 952215 *Tailor*.

References

[1] Furnell, S., Heyburn, H., Whitehead, A., Shah, J.N.: Understanding the full cost of cyber security breaches. Computer fraud & security **2020**(12), 6–12 (2020)

[2] Bada, M., Nurse, J.R.C.: The social and psychological impact of cyberattacks. In: Benson, V., Mcalaney, J. (eds.) Emerging Cyber Threats and Cognitive Vulnerabilities, pp. 73–92. Academic Press, USA (2020)

[3] Dinicu, A., Oancea, R., Bârsan, G.: The multidimensional impact on society of cyber attacks targeting the energy critical infrastructure sector. Land Forces Academy Review **26**(4), 406–417 (2021)

[4] Thakur, K., Ali, M.L., Jiang, N., Qiu, M.: Impact of cyber-attacks on critical infrastructure. In: 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 183–186 (2016). IEEE

[5] Macaulay, T.: Critical Infrastructure Interdependency. https://doi.org/10.5683/SP3/Y2CMPZ

[6] McLean, C., Lee, Y.T., Jain, S., Hutchings, C., Hurchings, C.: Modeling and simulation of critical infrastructure systems for homeland security applications. US Nat. Inst. Standard Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR **7785** (2011)

[7] Linkov, I., Baiardi, F., Florin, M.-V., Greer, S., Lambert, J.H., Pollock, M., Rickli, J.-M., Roslycky, L., Seager, T., Thorisson, H., Trump, B.D.: Applying resilience to hybrid threats. IEEE Security & Privacy **17**(5), 78–83 (2019)

[8] Bellini, E., Bagnoli, F., Caporuscio, M., Damiani, E., Flammini, F., Linkov, I., Liò, P., Marrone, S.: Resilience learning through self adaptation in digital twins of

human-cyber-physical systems. In: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 168–173 (2021)

[9] Roncone, G.e.a.: Apt44: Unearthing sandworm. Technical report (2024)

[10] Pederson, P., Dudenhoeffer, D., Hartley, S., Permann, M.: Critical infrastructure interdependency modeling: a survey of us and international research. Idaho National Laboratory **25**, 27 (2006)

[11] Applebaum, A., Baker, J., Beck, D., Haase, M.: Attack flows — beyond atomic behaviors. MITRE Engenuity (2022)

[12] Baiardi, F., Tonelli, F.: Twin based continuous patching to minimize cyber risk. European Journal for Security Research **6**, 211–227 (2021)

[13] Ryan, M.: Ransomware Revolution: The Rise of a Prodigious Cyber Threat. Springer, Berlin (2021)

[14] Strom, B., et al.: MITRE ATT&CK™: Design and philosophy (2020). https://www.mitre.org/news-insights/publication/mitre-attck-design-and-philosophy

[15] Quinonero-Candela, J., Sugiyama, M., Lawrence, N., Schwaighofer, A.: Dataset Shift in Machine Learning. MIT Press, USA (2009)

[16] Moreno-Torres, J., Raeder, T., Alaiz-Rodrguez, R., Chawla, N., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012)

[17] Aven, T.: On the meaning of a black swan in a risk context. Safety science **57**, 44–51 (2013)

[18] Paté-Cornell, E.: On "black swans" and "perfect storms": Risk analysis and management when statistics are not enough. Risk Analysis: An International Journal **32**(11), 1823–1833 (2012)

[19] Baiardi, F., Sgandurra, D.: Assessing ict risk through a monte carlo method. Environ. System and Decision (33), 1–14 (2013)

[20] Wang, C., Davies, J.: Formal model-driven engineering: Generating data and behavioral components. In: First Int. Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2012). EPTCS, vol. 105, pp. 100–117 (2012)

[21] Barricelli, B.R., Casiraghi, E., Fogli, D.: A survey on digital twin: Definitions, characteristics, applications, and design implications. IEEE access **7**, 167653–167671 (2019)

[22] Lehner, D., *et al.*: Digital twin platforms: Requirements, capabilities, and future prospects. IEEE Software **39**(2), 53–61 (2018)

[23] Tao, F., Zhang, H., Liu, A., Nee, A.: Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics **15**(4), 2405–2415 (2019)

[24] Qi, Q., Tao, F.: Digital twin and big data towards smart manufacturing and industry 4.0: 360-degree comparison. IEEE Access **6**, 3585–3593 (2018)

[25] The MITRE Corporation: CVE. https://cve.mitre.org/

[26] NIST: National Vulnerability Database. https://nvd.nist.gov/

[27] Carleton, J., Krishnamoorthi, S.: Automated Breach and Attack Simulation: The Cost and Risk Reduction Revolution is Here. Frost&Sulllivan (2020)

[28] Kennedy, D., O'Gorman, J., Kearns, D., Aharoni, M.: Metasploit: the Penetration Tester's Guide. No Starch Press, San Francisco (2011)