Security and Safety, Vol. 4, 2025003 (2025)

Security and Safety in Network Simulation and Evaluation

Research Article

Open Access

Information Network

Threat ripple model: A model to characterize business-oriented attacks based on business dependencies

Shiliang Ao[1,*], Binxing Fang[1], Xinguang Xiao[1,2], and Hongli Zhang[1]

[1] *School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China*
[2] *Antiy Technology Group Co.,Ltd, Beijing 100195, China*

Received: 24 December 2024 / Revised: 26 February 2025 / Accepted: 10 March 2025 / Published online: 28 April 2025

Abstract Traditional attack descriptions and threat modeling are discussed directly from the perspective of attacking infrastructure, *i.e.*, platforms, using malicious code. For example, it is believed that exploiting vulnerabilities to access the system, and then invading the target platform that support the specified business through lateral movement can achieve the purpose of attacking the business. The most classic Cyber Kill Chain model expresses the attack process almost directly as a life cycle of malicious code execution, but in fact there are many ways can be utilized by adversary, such as the dependencies among businesses. In this paper, we discuss threat transmission from a business perspective. In a business dependency sequence, if any of the businesses prior to the specified business is abnormal, it is unlikely that the business will operate normally either. This leads adversary to target various business support platforms of the business dependent sequence in order to disrupt the normal operation of the target business, rather than attacking through lateral movement. For adversary organizations whose goal is to paralyze the architecture which includes many systems, they will utilize the interrelationships of businesses in the architecture to make the effects of the attack transmit from business to business, this attack pattern cannot be described by traditional threat models. This paper constructs an architecture model that integrates the platform and business, and also constructs a threat model that reflects the ripple effect of threats utilizing the dependency among businesses. The threat model is able to characterize the logic of the transmission of the threat in the architecture after it encounters an attack. By using our architecture model and threat model to characterize real attack event and to model the financial scenario, this paper indicates that our threat modeling approach can be used for threat event assessment and threat effect inference.

Keywords Business dependency, Threat ripple, Architecture model, Threat model, Threat ripple model, Architecture security

Citation Ao S, Fang B, Xiao X and Zhang H. Threat ripple model: A model to characterize business-oriented attacks based on business dependencies. Security and Safety 2025; **4**: 2025003. https://doi.org/10.1051/sands/2025003

1 Introduction

There are two possibilities for a system to be anomalous as a result of being attacked: One is that it has been directly attacked, and the other, it has been indirectly affected by other associated system that is operating abnormally after being attacked. For the first situation, we are concerned about how to quickly repair the system and recover the business capacity it supports. This type of attack has been described

by many threat models [1–5]. However, for the second situation, a system may not have the opportunity to be directly attacked, but the possibility remains for it to be affected by the businesses of other systems on which it depends. Once an adversary discovers a business that can affect the target system and finds a vulnerable upstream business for it, attacking the business will cause system's abnormal operation become inevitable. For example, in the scenario of an airline architecture[1], it may be difficult for an adversary to launch a direct attack against the architecture and make it paralyzed. However, if the adversary is possible to attack the ticketing system and make it unable to recognize passengers' boarding documents, this will prevent a large number of passengers from boarding their flights by crowding the ticketing halls, resulting extensive delays in departing flights, and ultimately paralyzed the airline architecture.

While it is common knowledge that cross-architecture businesses have strong dependencies among different architectures, the pattern of attack based on dependencies among businesses has been long neglected, there are no models or analytical tools to discuss it, and existing threat models describing attacks on systems, are also unable to express it.

The effects that threats transmit and affect among businesses which caused by the pattern of attack. We refer to this pattern of affecting as "**Threat Ripple**". The gap of the discussion in the characterization of the threats to the systems needs to be filled from the perspective of business dependencies.

1.1 Why are dependencies among businesses ignored?

On the one hand, it is because we have long believed that a system is equivalent to a platform, and that if the platform is running well, the supporting business is functioning properly. On the contrary, if the platform runs abnormally, it will inevitably affect the business functions. Therefore, we often strip away business and focus only on whether the platform is being attacked, and all our efforts in system security are focused on platform security. This fact is confirmed by the large number of attacks which patterns are "attack-direct", *i.e.*, the attackers launch a scan to find vulnerable platforms, and then attack on the vulnerable points [6–11]. However, due to the long common understanding of conflating business security with general application security, we ignore the fact that attack the platform and attack the business are two different ways can both affect the operation of the system. For example, the nature of a DDoS attack is not to attack the platform, as its purpose is to consume the service capacity of the target. Although the effect of DDoS attacks on the business level has been widely reported, it is still customarily to categorize them as attacks on the platform.

On the other hand, in threat scenarios, even though some of the systems' businesses are out of service, we usually don't consider them as the real target of the adversary because those platforms are not running abnormally. For some adversaries, their intention is to disrupt upstream businesses through the platforms, thereby affecting downstream targeted business in the business dependency sequence, so ostensibly the compromised platforms themselves did not appear to be operating abnormally. If we concern ourselves with attacks aimed at paralyzing the business, then we will focus on the business dependencies. For some APT organizations, the difference between these adversaries and the attackers is that the adversaries' attack mode is "direct-attack". They will utilize any means to achieve their goal of paralyzing the business. When it is impossible to attack the target's support platform (*e.g.*, the platform is network and physically isolated) or the other platforms which it depends (*e.g.*, isolated intranet host supporting Iranian nuclear weapons integration operates independently of other platforms), they usually attack the upstream business to affect the target business through "business-to-business" dependencies [12–16].

For example, in version 1.x of the Stuxnet virus, the adversary inserted into the PLCs which used to control the centrifuges was not to disrupt their original functions, but affect the downstream businesses through them. The adversary modified centrifuges operating data and led to their abnormal speeds, so the downstream uranium separation business could not meet the requirements of weapons production–the adversary achieved the goal of stagnating the development of nuclear weapons [13, 14, 16]. Besides this, the Ukrainian blackout event was also a case of a cyber-attack affecting a non-cyber business [15].

[1] In our paper, an architecture is identified as a collection of systems. While the systems involved are independent on management and/or operation, their interoperable and/or integrated combination could often accomplish missions that could not be done by a single system.

These attacks, which seem to be on the platform but are actually on the business, give us an inspiration: it is not necessary to attack the support platform to affect a business; if the business belongs to a business dependency sequence, the adversary makes its upstream business operate abnormally, it will also cause the downstream target business to become abnormal. Therefore, only by solving the business-oriented attack description and expressing the effects brought by the business being attacked, we could accurately characterize the threat ripple effects even more and construct a complete architecture threat model.

1.2 How to characterize business-oriented attacks?

In order to express the threat events more accurately, we decompose the system into "platform + business" from the adversary's intention, in order to characterize the two attack modes of "attack-direct" and "direct-attack" from the vulnerability level of the platform and the dependency level of the business. We construct a new architecture model to express the information interactive relationship among the platforms and the businesses. Also, we construct a threat ripple model combined with the architecture model, to characterize the transmissions and attribute changes of threats among the associated systems in the architecture after the architecture being attacked. A specific business may have difficulty to find a direct path to attack it, but it can be affected by other businesses it depends, which can result in the specific business being affected or even paralyzed by the associated effects.

Our threat ripple model not only take into account the focus on attacks against platforms and businesses in a traditional sense, but also the ripple effects of threats within business-to-platform and business-to-business which been long neglected, and characterize the entire effects of attack on an architecture accurately and completely.

Our model can also effectively express the logic of attack that aiming cross-modal architecture which has completely different attributes from the target architecture but has a strong relationship with, in order to affect the target architecture, *e.g.*, via launching attacks on businesses outside the target architecture through social network attacks, social engineering, supply cutoffs, *etc.*, utilizing the threat ripple effects to affect businesses inside the target architecture and make them operate abnormally. This provides a theoretical basis for characterizing the logic of threat transmissions among cross-modal businesses.

Our contributions:

- **Construct an architecture model that integrates the platform and the business.** The model provides a perspective that can directly observe the business, while the normal operating of the business is the most fundamental security factor of a system, and the purpose of attacking the platform is essentially to attack the business. In this paper, we construct an architecture model that can characterize the information interactive relationship among the platforms and the businesses, which can help to build endogenous security in the process of architecture design, operation and maintenance, also discover the transmission paths that enable risks and threats to spread the effects, in order to settle the potential security problems existing in the current information architecture in advance.
- **Construct a threat model that reflects the ripple effect of threats utilizing the dependencies among businesses.** This paper proposes a threat model from a business perspective that can characterize the propagation of threats utilizing dependencies among businesses. It can express the logic of the transmission and of threats in the architecture after the architecture being attacked. This helps defenders quantify the business effects of the threats as they respond and assess the threat events while encountered. Actually, our threat model can also characterize the logic of generate ripple effects by disrupt or interfere the business support elements, *e.g.*, operational crew, physical resources, and operating capital, of the target architecture.
- **Propose a methodology to discover and determine the vulnerable points of business and platforms in the architecture.** For the business logic in the architecture, our threat ripple model is able to determine which platform or business could affect the important business once they are attacked. This helps the defender to infer which nodes are the adversary's key targets, thus determine the security guarantee nodes for business. Defenders could characterize their architecture according to our model, then infer attacks through the threat model, and finally locate platform and business vulnerabilities directly. Besides security assessment, through building threat ripple model, defender is able to stop new effects in time that might transmit from the threat generated by attack, even if

the attack has already occurred. In addition, this paper could also guide our attention to vulnerable business and platform elements in other cross-modal architectures, which could affect the current architecture through threat ripple if they are attacked.

The remainder of this paper is organized as follows. In Section 2, related work about threat models are discussed. In Section 3, the architecture model is constructed to characterize the dependencies among platforms and business. Section 4 introduces the definition of threat ripple. In Section 5, the threat ripple model is constructed based on the architecture model. Section 6 characterizes a typical cyberattack threat event utilizing the threat ripple model. Section 7 introduces how to make attack inference through our models. Finally, Sections 8 and 9 provide the discussion on future research directions and conclusion of this study, respectively.

2 Related work

From an adversary's and a defender's perspective, we divide threat models into two categories.

2.1 Adversary-oriented threat model

The most representative cyber kill chain model divides the life cycle of an attack into seven phases [17]. Besides, SANS Malware Forensics Kill Chain [18], ArcSight Attack Life Cycle [19], United States DoD Cybersecurity Kill Chain [20], Industrial Control System Cyber Kill Chain [21], Unified Kill Chain [22] are also typical kill chain models. Although they can be used to characterize the process of an adversary's attack on an asset, they lack the coverage of the threat transmission process among businesses after the attack.

For threat models oriented to attack behaviors and actions, the MITRE's Adversary Attack, Techniques & Common Knowledge (ATT&CK) is an adversary model and framework for describing the actions an adversary could take to compromise and operate within an organization network [23]. NIST SP 800-30 list of threat events [1], CAPEC [24], MITRE's TARA [25] provide resources of threat modeling from tactics, techniques, and procedures-oriented dimensions (TTP), greatly improves the ability of the defender to investigate the adversary. But these models express the attack exclusively as an attack action on the platform, and none of them consider the attack on the business.

2.2 Architecture-oriented threat model

Previous studies have constructed typical generic frameworks and models that could be applied in large information system scenarios including ONDI Cyber Threat Framework [26], Cyber Prep 2.0/DACS [27], Attack Tree Model [3]. These frameworks and models provide clear approaches to describe the threats facing cyber asset systems. In threat and risk assessment, OWASP [28], PASTA [29] and ADVISE [30] provide modeling methods orienting technology to assess the risks to the system. Models that support design analysis and test including STRIDE & DREAD [5], NIST SP 800-154 [4], and OCTAVE [2] can help systems to assess their security. But these models are constructed from the perspective of the platform.

Structured Threat Information eXpression (STIX) uses adversary TTPs, attack events, process of actions, targets to exploit, threat actors, and other methods to provide a common mechanism for adding structured cyber threat intelligence information across a range of use cases for improving consistency, efficiency, interoperability, and overall situational awareness [31]. In terms of the secure design of the system, for large systems or architectures with complex integration and interoperability challenges, TOGAF [32] and DODAF [33] provide approaches to designing, planning, implementing, and managing enterprise information and technology frameworks. The two models are generally used to model at four levels including business, application program, data, and technology. Furthermore, DODAF also supplies operation diagrams to provide visualized information of base structure for issues concerned by specific stakeholders. However, neither these models nor security design approaches to the systems or architectures take business security into account.

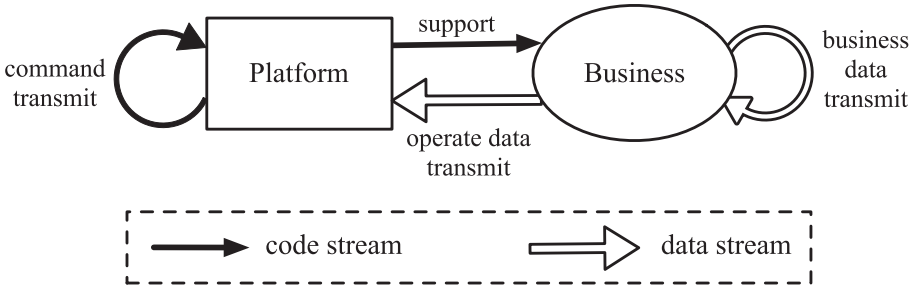

Figure 1. Architecture model

3 Architecture model

We constructed an architecture model, shown in Figure 1, to express the mission processing pattern of the information architecture and the relationship between the platform and the business:

Platform refers to the infrastructure which support the business operation through running code, expressed as a solid rectangle.

Business refers to the service functions that the system provides to the outside world through data, expressed as a solid ellipse.

The information interactions between them have been labelled in Figure 1, where we have used single line arrows to express the code stream and double line arrows to express the data stream. There are four corresponding binary relations:

Command transmit: Platform → Platform. Commands are transferred from platform to platform through code stream to achieve functional synergy among platforms.

Business Data transmit: Business → Business. Information is transferred from business to business through business data stream to achieve capability synergy among businesses.

Support: Platform → Business. The platform supports its corresponding business through running code. Once the platform is down, the corresponding business naturally disappears.

Operate Data transmit: Business → Platform. Business uses its corresponding platform through entering operation data into the platform. Note that for a platform, the business which inputs operate data to, and its supporting business, are not necessarily to be the same.

Based on the operation and management patterns of the architecture, the operators can characterize the business processes that correspond to each mission, as well as the involved interactions, data flows and capability dependency information among the platforms and the businesses, with timely updates as the architecture evolves and adapts.

4 Threat ripple

We define the new objects added to the following definitions as follows: the threat source is the adversary (including attacker) who initiates the threat event, or the agent that can be used to initiate the attack, expressed as a circle. We use single line arrows to express the path of the threat source using code stream to attack and affect the platform, and double line arrows to express the path of using data stream to attack and affect the business.

Predefinition 1: Attack platform

The traditional pattern of attacks on platforms is that an adversary who successfully attacks a platform will use it as a springboard to move on to the next platform. The attack path from the initial platform to the target platform can be used to describe the adversary's specific attack behavior. Therefore, the attack on these platforms is a sequence of network nodes, denoted as $Q_p = [P_1, P_2, \ldots, P_n]$, as shown in Figure 2a. And current cyberattacks in actual fact make threats to be transmitted between platforms and ultimately have an effect. The process of adversary launching a cyberattack is, in fact, the process threat transmitting among platforms and finally causing effects just like the definitions of cyberattack and cyberattack lifecycle:

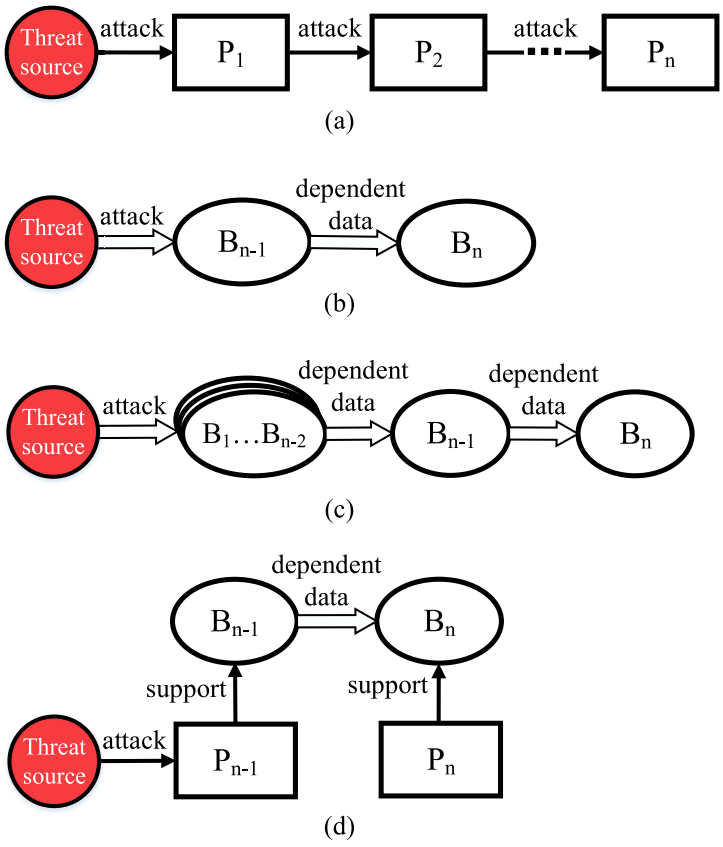

Figure 2. Attacks launched by the adversary. (a) Attack Platform. (b) Attack target business through dependent business. (c) Attack target business through business dependency sequence. (d) Attack support platform to affect target business

Cyberattack. An attack, via cyberspace, targets an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information [34].

Cyberattack lifecycle. Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Actions on Objective [17].

The current definition of cyberattack has been essentially limited to the attacks on platforms, so do the cyberattack lifecycle was also limited to the process of spreading the threat among platforms. What happened among platforms and businesses were neglected in both definitions above.

Predefinition 2: Attack Business

There are dependencies between businesses in an architecture, *i.e.*, the normal operation of a business depends on the preceding business provides with normal data, and we say that the operational capability of that business depends on the preceding business. Therefore, for business B_n, its business operation capability depends on the business B_{n-1} in front of it, and attacking business B_{n-1} will lead to abnormal operation of business B_n as shown in Figure 2b.

For business dependency sequence $Q_B = [B_1, B_2, \ldots, B_n]$, interrelationships f_{supply} exist between businesses:

$$f_{supply}: \ B_{i-1} \rightarrow B_i, i \in [1, n]$$

i.e., for the business B_n in the sequence to operate normally, it is necessary for any of the businesses in front of it to be able to provide normal dependent data to the one behind it, as shown in Figure 2c.

If any of the business in Q_B is attacked and operates abnormally, it will lead to its inability to provide normal dependent data to its subsequent business, which will eventually and inevitably lead to the abnormal operation of B_n. Therefore, we have the definition of attack business:

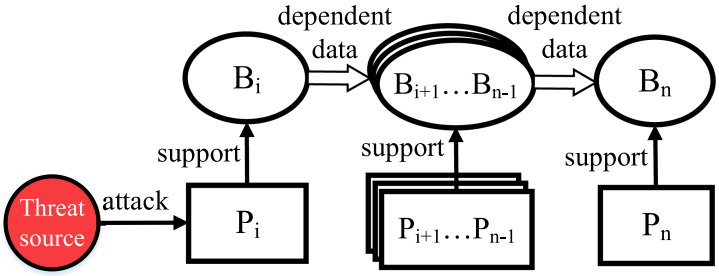

Figure 3. Attack a support platform in business dependency sequence to affect the target business

Attack business. If the adversary's goal is to make business B_n operate abnormally, then attack any of the business in its business dependency sequence Q_B will achieve the goal.

Predefinition 3: Utilizing relationships between platform and business to execute attacks

Since each business requires a support platform for its normal operation, if the adversary's goal is to complete an attack on B_n, an attack on P_{n-1} will also be able to achieve the goal of attacking B_n. This is because a successful attack on P_{n-1} results in abnormal operation of B_{n-1}, which leads to abnormal operation of B_n, as shown in Figure 2d.

We denote the support platform required for the normal operation of any one of the business B_i in the business dependency sequence Q_B as P_i[2], there is a interrelationship $f_{support}$ between them:

$$f_{support} : P_i \rightarrow B_i, i \in [1, n]$$

i.e., an adversary launching an attack on P_i which is the support platform for any of the business in Q_B, is able to complete an attack on B_n. Because once P_i is running abnormally, it will inevitably result in losing support and operating abnormally of B_i, which will cause the threat to be transmitted in Q_B and ultimately lead to the B_n operating abnormally. as shown in Figure 3.

Therefore, we have the definition of attack business support platform:

Attack business support platform to affect business. If the adversary's goal is to make business B_n operate abnormally, launch an attack on its support platform P_n, or launch an attack on the support platform P_i of any business B_i in the business dependency sequence, will be able to achieve this goal.

Definition of threat ripple:

We combine the above predefinitions and generalize the way of attacking the business to get the definition of threat ripple:

Threat ripple. When the operation of target business depends on the operation of its preceding businesses, and they form a business dependency sequence, then attacking any business in the sequence (including target business), or attacking the platform that support it, will lead to abnormal operation of the business and all its subsequent businesses, ultimately make the target business operation abnormal. We define this pattern of threat transmission among businesses as threat ripple.

The pattern of utilizing threat ripples to attack the target business is applied to attack information architectures with complex business dependencies. Typically, during the design, construction and management of the architecture, the business coordinations and processing processes that execute to fulfil the missions, as well as the specific support platforms required for the operations, have already been determined. The adversary can obtain this information through intelligence gathering, network scanning, *etc.*, and then explore the vulnerable objects and develop attack plans.

[2] P_i can be a standalone platform or a combination of platforms that support the business B_i, which we logically see as a whole. If any part of the combination is attacked and a problem occurs resulting in P_i running abnormally, the combination is considered to be abnormal. If it does not result in a problem, P_i is considered to be tolerant of intrusion against this attack, and the threat will not spread.

5 Threat ripple model

In this paper, a threat ripple model is constructed based on the architecture model, which is used to express the logic of threat transmission within the architecture after the architecture been attacked.

The **normal state** of a platform and business is the state in which it is able to perform its intended mission correctly and safely, and the **abnormal state** of a platform and business is a state in which its resources, functions, privileges, operation or running, data or communications are suffered in such a way that it is unable to perform its intended mission correctly and securely. When threats are encountered, the transition from a normal state to an abnormal state, is results from the platform or business operating in order to complete the mission after receiving unanticipated inputs. We abstract the actions of threat transmission that lead to changes in the operational state of the platform and the business into two types: **attack** and **affect**.

For platform, the **attack** is defined as changing the running state of the platform through code with attack attributes.

For business, the **attack** is defined as the action of changing the operational state of the business through data with attack attributes.

The **affect** is defined as without any new attack code or attack data, but due to its own running state or operational state being affected, platform follows the normal logic of interacting with associated platform in code will cause the associated platform's running state to change, as well as business follows the normal logic of interacting with associated business in data will cause the associated business's operational state to change.

Threats can arise after the platform and business being attacked or affected, and can be transmitted through data streams or code streams.

5.1 Objects definition

In the model, we define the newly added objects as follows:

Affected Platform is defined as a platform that suffers an attack or effect and results in an abnormality in the running state of the platform and is able to continue spreading its effects outwards and affecting other platforms or business, expressed as a dotted rectangle. After being attacked and compromised, then it can be the adversary's new threat agent and will be able to launch another attack.

Affected Business is defined as a business that suffers an attack or effect and results in an abnormality in the operating state of the business and is able to continue spreading its effects outwards and affecting other businesses or platforms, expressed as a dotted ellipse.

The presentation of the dotted lines in the figures means that they are no longer secure and that they will continually spread threats outwards, thus becoming new threat transmitters.

If the platform or the business was attacked or affected, even though the outcome affects the platform's operation status and the business's operation status, these threats are no longer spreading, or the consequences of the threats are within the intrusion tolerance of platform's running or business's operations, then their markings will remain and they are still expressed as solid rectangles and solid ellipses. The presentation of the solid lines in the figures means that although they are affected, the threat does not spread and they do not become new threat transmitters.

5.2 Relationship definitions

We provide a formal representation of the relationships between objects in the threat ripple model. We define the threat ripple model L:

$$L = \{S, F, G\}$$

where S is defined as the set of all objects in the model L, F is the set of attack functions, and G is the set of affect functions. For S:

$$S = \{\mathit{Threat\ Source}, \mathit{Affected\ Platform}, \mathit{Affected\ Business}, \mathit{Platform}, \mathit{Business}\}$$

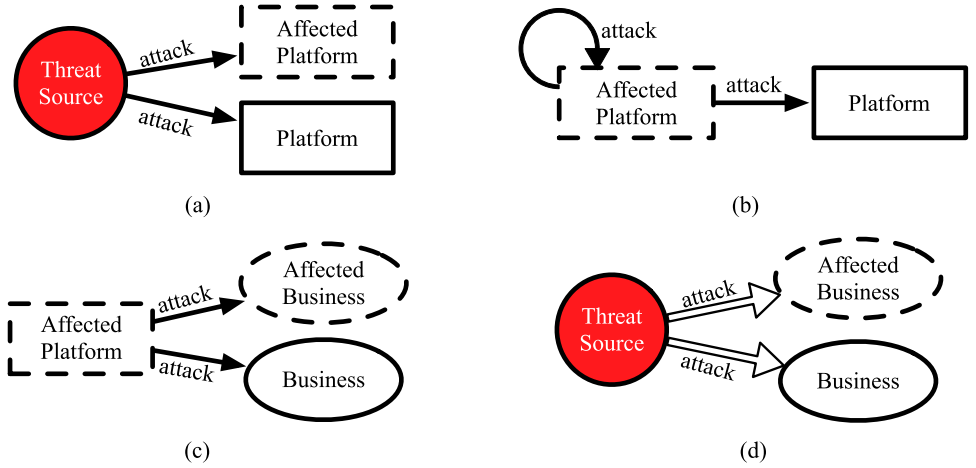

Figure 4. Attack mapping. (a), (b) and (c) are the mapping expansion of $f_{attack\ code}$. (d) is the mapping expansion of $f_{attack\ data}$

For the objects of the threat's action, there exists subsets: S_k, S_t, S_p, $S_b \subset S$, where:
S_k is the set of objects that launch the attack, *i.e.*:

$$S_k = \{Threat\ \ Source, Affected\ \ Platform\}$$

S_t is the set of objects that transmit the effect, *i.e.*:

$$S_t = \{Affected\ \ Platform, Affected\ \ Business\}$$

S_p is the set of platform objects that threats act on, *i.e.*:

$$S_p = \{Affected\ \ Platform, Platform\}$$

S_b is the set of business objects that threats act on, *i.e.*:

$$S_b = \{Affected\ \ Business, Business\}$$

For the threat actions among the objects of S_p, S_b in L, we define them into two types of mapping relationships: **attack** mapping relationships and **affect** mapping relationships.

5.2.1 Attack mapping function

The set F contains two mapping functions used to describe eight **attack** mapping relationships from S_k to S_p and S_b:

$$F = \{f_{attack\ code}, f_{attack\ data}\}$$

Where $f_{attack\ code}$ represents the attack executed through code stream. For the objects in S_k attack via code stream, thereafter make effect on the objects in S_p, the mapping relationship is expressed as:

$$f_{attack\ code}:\ S_k \rightarrow S_p \tag{1}$$

We expand Equation (1), the two **attack** mapping relationships initiated by threat source through code stream are shown in Figure 4a:

$\boldsymbol{f_{attack\ code}}$**: Threat source → affected platform.** The adversary launches an attack on the platform through code stream, causing it to run in an abnormal state, and the threat will continue to spread. *e.g.*, in the Ukraine blackout event, the adversary took control of the SCADA system via the BlackEnergy Trojan. Then, the adversary gave the command to cut off the power [15, 35].

$\boldsymbol{f_{attack\ code}}$**: Threat Source → Platform.** The adversary launches an attack on the platform through code stream, causing it to run in an abnormal state, but the threat will no longer spread.

e.g., the report "Mobile Malware Threat Landscape 2022" released by Kaspersky states that attackers will use Trojans to hack into the user's mobile terminal and upload the user's data. At this time, the mobile terminal can still run normally, but the user's data has been stolen [36].

The two **attack** mapping relationships initiated by affected platform through code stream are shown in Figure 4b:

$\boldsymbol{f_{attack\ code}}$**: Affected Platform → Affected Platform.** The attack agent launches an attack on the platform through code stream, causing it to run in an abnormal state, and the threat will continue to spread. *e.g.*, in the Stuxnet event, after compromising an industrial network user terminal, the adversary used it as a springboard to move laterally to attack other hosts [13, 16].

$\boldsymbol{f_{attack\ code}}$**: Affected Platform → Platform.** The attack agent launches an attack on the platform through code stream, causing it to run in an abnormal state, but the threat will no longer spread. *e.g.*, in the attack event of SWIFT service provider EastNet, the adversary used the compromised server in the MGMT region as a springboard to compromise an Oracle database server in the SAA region, then steal its data. Although the server remained operational, the data was stolen [12].

For the affected platform attack the objects in S_b via running code[3], the mapping relationship is expressed as:

$$f_{attack\ code}: \ Affected\ Platform \rightarrow S_b \tag{2}$$

We expand Equation (2), the two **attack** mapping relationships initiated by affected platform through running code are shown in Figure 4c:

$\boldsymbol{f_{attack\ code}}$**: Affected Platform → Affected Business.** The attack agent launches an attack on the business through running code, causing it to operate in an abnormal state, and the threat will continue to spread. *e.g.*, in the event of WannaCry, the worm, after attacking users' Windows computers, would ransom them and ask for payment via Bitcoin. This stagnates the work tasks of current users and prevents the operation of any of the businesses involved [11].

$\boldsymbol{f_{attack\ code}}$**: Affected Platform → Business.** The attack agent launches an attack on the business through running code, causing it to operate in an abnormal state, but the threat will no longer spread. *e.g.*, in the Heartbleed vulnerability analysis report, an attacker who compromised an intranet host was able to send maliciously constructed heartbeat packets to an OpenSSL server with the vulnerability to remotely access 64K of data in memory that had crossed the boundary. The server's supporting business can still operate normally, but data has been leaked [9].

Where $f_{attack\ data}$ represents the attack executed through data stream. For the threat source attack via data stream, thereafter make effect on the objects in S_b, the mapping relationship is expressed as:

$$f_{attack\ data}: \ Threat\ Souece \rightarrow S_b \tag{3}$$

We expand Equation (3), The two **attack** mapping relationships initiated by threat source through data stream are shown in Figure 4d:

$\boldsymbol{f_{attack\ data}}$**: Threat Source → Affected Business.** The adversary launches an attack on the business through data stream, causing it to operate in an abnormal state, and the threat will continue to spread. *e.g.*, in the Ukraine blackout event, the adversary launched a telephony DDoS attack against the electricity customer service center, causing customer service business breakdown. This prevented crews from determining the area of the outage and from taking measures to deal with it [15, 35].

$\boldsymbol{f_{attack\ data}}$**: Threat Source → Business.** The adversary launches an attack on the business through data stream, causing it to operate in an abnormal state, but the threat will no longer spread. *e.g.*, with Bitcoin soaring in 2019, attackers began using DDoS attacks to ransom some companies' websites, preventing them from responding normal access. The threat from the attackers did not spread, but companies were forced to disrupt their web access business [6].

[3] The platform runs code to affect the supported business, and in fact it is the running result that causes the business to be abnormal. This paper views this type of condition as a special form of code stream sent from the platform to the business.

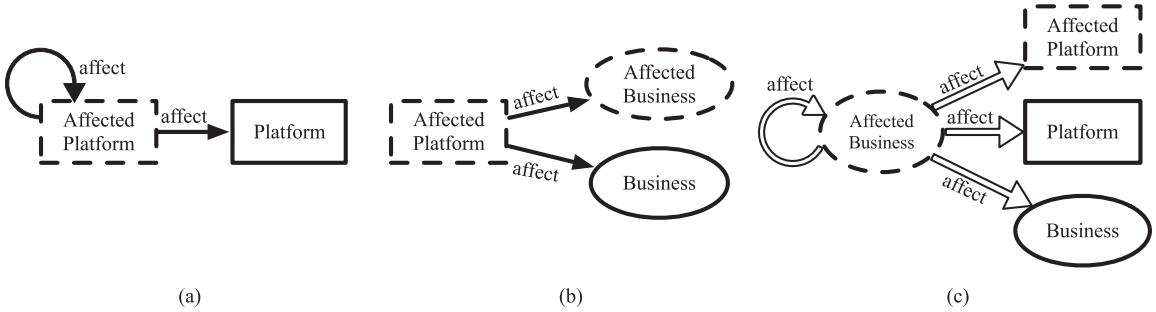

Figure 5. Affect mapping. (a) and (b) are the mapping expansion of $g_{affect\ code}$. (c) is the mapping expansion of $g_{affect\ data}$

5.2.2 Affect mapping function

The set G contains two mapping functions to describe eight **affect** mapping relationships from S_t to S_p and S_b:

$$G = \{g_{affect\ code},\ g_{affect\ data}\}$$

Where $g_{affect\ code}$ represents the affect through code stream. For the affected platform affect the objects in S_p and S_b via code streams, the mapping relationship is expressed as:

$$g_{affect\ code}:\ Affected\ Platform \rightarrow S_p, S_b \tag{4}$$

We expand Equation (4), the two **affect** mapping relationships where threats transmit through code streams initiated by affected platform are shown in Figure 5a:

$\boldsymbol{g_{affect\ code}}$**: Affected Platform → Affected Platform.** The affected platform transmits threat to other platform through code stream, causing it to run in an abnormal state, and the threat will continue to spread. *e.g.*, in the attack event on SolarWinds, the attacker replaced the Orion software source code provided on its official website with a backdoor-implanted version, thus implementing a supply-chain prefabricated attack. When users normally visit the website then download and use the installation package, the attacker can control the user's host through the backdoor, and use it as a springboard to move laterally to attack other hosts [8, 37].

$\boldsymbol{g_{affect\ code}}$**: Affected Platform → Platform.** The affected platform transmits threat to other platform through code stream, causing it to run in an abnormal state, but the threat will no longer spread. *e.g.*, in the attack event on SolarWinds, the attacker compromised the distribution environment of the company's Orion infrastructure management platform and added a backdoor to the source code. These legally digitally signed codes were distributed to the users along with the software updates. When user run the updated software, the attacker could steal data from user through the backdoor. This time, the user's software could still run normally, but it was already in a state of intrusion [8, 37].

The two **affect** mapping relationships where threats transmit through running code initiated by affected platform are shown in Figure 5b:

$\boldsymbol{g_{affect\ code}}$**: Affected Platform → Affected Business.** The affected platform transmits threat to other business through running code, causing it to operate in an abnormal state, and the threat will continue to spread. *e.g.*, in the Stuxnet event, after compromising the control host of the SCADA system, the Stuxnet worm modified the operating data in the valve PLC controllers of the cascade protection systems and the speed PLC controllers of the Iranian uranium centrifuges through data streams. This resulted in the uranium centrifuge business operating with incorrect data [13, 16].

$\boldsymbol{g_{affect\ code}}$**: Affected Platform → Business.** The affected platform transmits threat to other business through running code, causing it to operate in an abnormal state, but the threat will no longer spread. *e.g.*, an attacker maliciously modifies a company's promotional website by rewriting the inquiry link as a link to the attacker's own spoofed webpage. The corresponding business on the company's website continues to operate normally, but users will mistakenly visit the attacker's web page and obtain fraudulent information that the company is about to be closed down [7].

Where $g_{affect\ data}$ represents the affect through data stream. For the affected business affect the objects in S_p and S_b via data streams, the mapping relationship is expressed as:

$$g_{affect\ data}:\ Affected\ Business \rightarrow S_p, S_b \tag{5}$$

We expand Equation (5), the four **affect** mapping relationships where threats transmit through data stream initiated by affected business are shown in Figure 5c:

$\boldsymbol{g_{affect\ data}}$**: Affected Business → Affected Platform.** The affected business transmits threat to other platform through data stream, causing it to run in an abnormal state, and threat will continue to spread. *e.g.*, in the XcodeGhost event, the software company developed software using the polluted Xcode editor, which resulted in the software being polluted. Then the software was uploaded to online locations such as web cloud disks and forums. The software, when downloaded and installed by the user, would open a remote-control entry for the attacker. Attackers could enter the host and implant new malicious code, in order to further launch the attack [10].

$\boldsymbol{g_{affect\ data}}$**: Affected Business → Platform.** The affected business transmits threat to other platform through data stream, causing it to run in an abnormal state, but the threat will no longer spread. *e.g.*, in the XcodeGhost event, the software company used the polluted Xcode editor to develop an APP and uploaded it to Apple's APP Store, where it passed the review. After the user installed the polluted software, the user's mobile phone data would be sent back to the attacker's specified domain. By this time, the users' data has been leaked, but the threat will no longer spread [10, 38].

$\boldsymbol{g_{affect\ data}}$**: Affected Business → Affected Business.** The affected business transmits threat to other business through data stream, causing it to operate in an abnormal state, and threat will continue to spread. *e.g.*, in the Stuxnet event, after the Stuxnet worm modified the speed data of the PLC controllers for centrifuge speed at Iran's industry of control system (ICS), some centrifuges in the uranium centrifuge business were damaged as a result of constantly variable-speed operation. This caused reduced capacity of the uranium production [13, 16].

$\boldsymbol{g_{affect\ data}}$**: Affected Business → Business.** The affected business transmits threat to other business through data stream, causing it to operate in an abnormal state, but the threat will no longer spread. *e.g.*, in the Stuxnet event, although the functions of the nuclear weapon integration business were normal, the lack of uranium prevented it from producing nuclear weapons due to the lack of capacity in the uranium production business [13].

The above mapping relationships express the pattern of the threat transmission set in model L, where Equation (1), Equations (2) and (4) are the transmission of threat through code stream and Equations (3) and (5) are the transmission of threat through data stream. For an attack event in a real scenario, the attack and affect processes it contains are usually a combination of some mapping relationships above.

5.3 Model representation

In this section, we construct the ripple model $L = \{S, F, G\}$. Where S is the set of objects in the model, including threat source, affected platform, platform, affected business and business.

F is the set of attack functions with three attack mapping relationships:

$$\begin{aligned} F = \{&S_k \rightarrow S_p, \\ &Threat\ Source \rightarrow S_b, \\ &Affected\ Platform \rightarrow S_b\} \end{aligned} \tag{6}$$

G is the set of affect functions with two affect mapping relationships:

$$\begin{aligned} G = \{&Affected\ Platform \rightarrow S_p, S_b, \\ &Affected\ Business \rightarrow S_p, S_b\} \end{aligned} \tag{7}$$

Our model is able to not only characterize the traditional attacker attacks on platforms and businesses, but also the affect based on the relationships of platform-to-platform, business-to-platform, and business-to-business. These express the complete ripple logics of the threat within the architecture. The threat ripple model is shown in Figure 6.

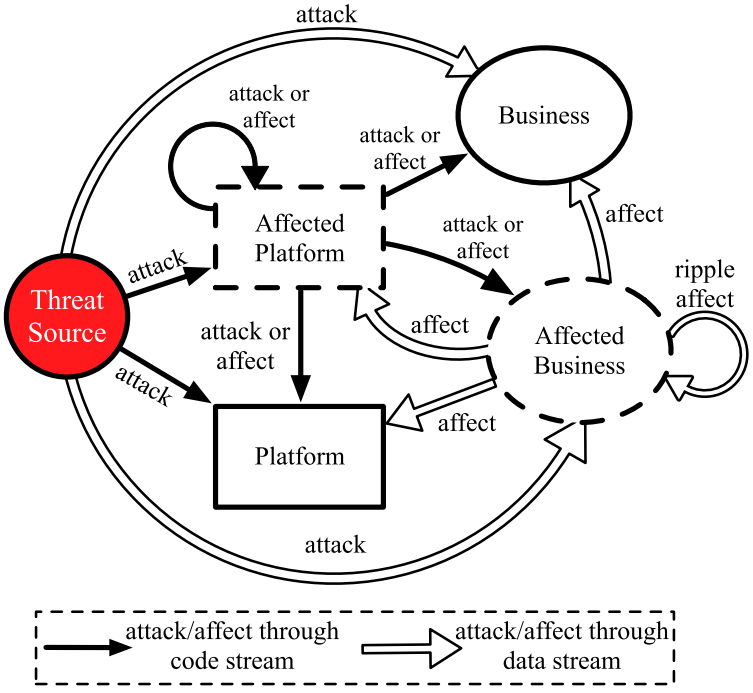

Figure 6. Threat ripple model

5.4 Mathematical representation

The mathematical representations are concisely constructed for the threat ripple model to enable it to support algorithmic applications that allow automated analysis and inference for threat ripples. The objects in architecture are defined as follows:

$$B = \{B_1, B_2, \ldots, B_t\}$$

where B is the set of businesses and the support platform for business $B_i \in B$ is P_i, the set of support platforms is:

$$P = \{P_1, P_2, \ldots, P_t\}$$

The business that directly depends on business B_i is denoted by $B_j(i)$, and the platform that supports it is denoted by $P_j(i)$. The threat source is denoted by T.

For business B_i and platform P_i, their abnormal states are denoted by $S^F_{B_i}$ and $S^F_{P_i}$ after suffering threats, respectively. Their states are expressed as $\tilde{S}_{B_i}$ and $\tilde{S}_{P_i}$ if they continue to transmit threats, and as $\hat{S}_{B_i}$ and $\hat{S}_{P_i}$ if they do not transmit threats, *i.e.*:

$$S^F_{B_i} \in \{\tilde{S}_{B_i}, \hat{S}_{B_i}\}$$

$$S^F_{P_i} \in \{\tilde{S}_{P_i}, \hat{S}_{P_i}\}$$

Where T and $\tilde{S}_{P_i}$ can launch an attack, we denoted them by A_K, *i.e.*:

$$A_K \in \{T, \tilde{S}_{P_i}\}$$

For the attack mapping functions in Section 5.2.1, we use f^C and f^B to express $f_{attack\ code}$ and $f_{attack\ data}$, respectively. Then for Equation (6), the mathematical expressions of the attack functions in set F are:

$$\begin{aligned} F = \{&f^C(A_K) = S^F_{P_j(i)}, \\ &f^B(T) = S^F_{B_j(i)}, \\ &f^C(\tilde{S}_{P_i}) = S^F_{B_j(i)}\} \end{aligned}$$

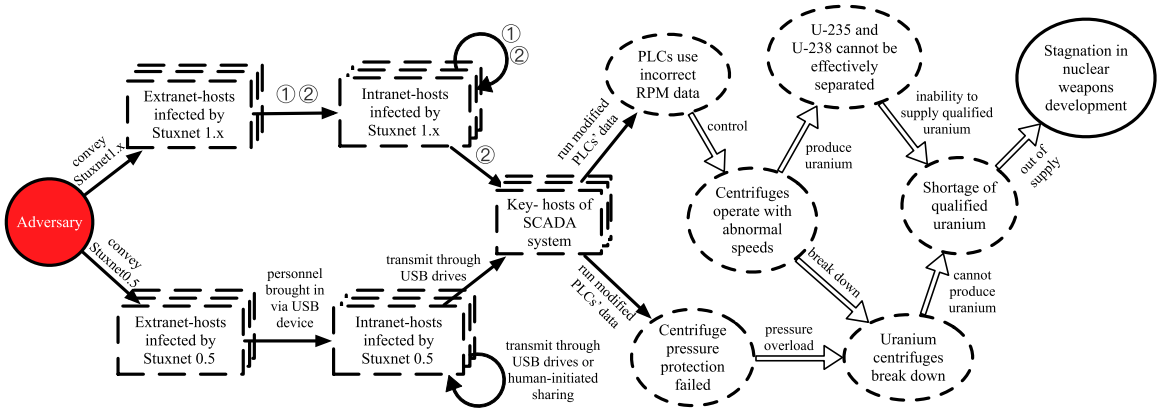

Figure 7. Threat ripple characterization of Stuxnet event. ① represents the network transmission mode, and ② represents the USB mobile media transmission mode, as described in Section 6

For the affect mapping functions in Section 5.2.2, we use g^C and g^B to express $g_{affect\ code}$ and $g_{affect\ data}$, respectively. Then for Equation (7), the mathematical expressions of the affect functions in set G are:

$$G = \{g^C(\tilde{S}_{P_i}) = S^F_{P_j(i)}, S^F_{B_j(i)}, \\ g^B(\tilde{S}_{B_i}) = S^F_{P_j(i)}, S^F_{B_j(i)}\}$$

The mathematical expression of threat ripple is a complex theory, the specific forms and properties must be strictly defined and interpreted in detail. Due to the structure and length of this paper, the above is only a brief description[4].

6 Characterize a typical cyberattack event

We map realistic Stuxnet attack event [13, 14, 16] to the threat ripple model, as shown in Figure 7. The Stuxnet event, an attack aimed at stagnating Iran's nuclear weapons development process, is sufficient proof that cyberattacks can play an important role in military operations. The worms are divided into Stuxnet 0.5 [16] and Stuxnet 1.x [14, 16] versions, their final goal is to disrupt uranium centrifuge infrastructure by modifying parameter data of programmable logic controllers (PLCs) used to configure the industrial control system (ICS).

After Stuxnet 0.5 was conveyed, it first infected extranet hosts of the enterprise network, which in turn infected inserted-in USB devices, and then these devices contained Stuxnet 0.5 virus was taken into the target enterprise network through personnel-brought-in manner. When inserted into a new host in the enterprise network, Stunex 0.5 infects the Siemens Step 7 project files in the host. These infected files will then be spread among hosts on the enterprise intranet via USB devices or human file sharing (*e.g.*, via email), and a P2P network will be set up among infected machines to enable sharing of updates to the malicious code via mailslots. When Stuxnet 0.5 finally succeed in spreading to and infecting key hosts in the SCADA control system network, it would implant malicious code by replacing legitimate DLL files of Siemens' proprietary Step 7 control software, which in turn modified the PLC parameter data to pressurize the centrifuge. This resulted in the centrifuges operated under overpressure conditions and were ultimately damaged.

After Stuxnet 0.5 executed the infection, the adversary organization conveyed multiple new versions of the worms used to damage centrifuges to the enterprise network, which were collectively referred as Stuxnet version 1.x. They spread via two modes: network and USB removable devices, as shown in Figure 7.

For the mode of propagation through the network ①, there are four ways to infect new hosts:

[4] The detailed and strict mathematical descriptions for threat ripple are presented in "The RipA Model: On the Domino Effect in Attacks against Informatization Architectures".

- Infecting remote hosts via MS10-061 Print Spooler Zero-Day Vulnerability.
- Infecting remote hosts via MS08-067 Windows Server Service Vulnerability.
- Accessing and infecting hosts running WinCC database software by utilizing hard-coded database server passwords.
- Spreading via Network share utilizing scheduled job or Windows Management Instrumentation (WMI).

For the mode of propagation via USB removable media ②, Stuxnet 1.x replicated itself to inserted removable drives and infects new hosts in 2 ways:

- Replicating itself and infecting via exploiting the MS10-046 shortcut LNK vulnerability that allows auto-execution when viewing removable drives.
- Utilizing the autorun command to specify the malicious file to be executed as the actual autorun.inf file so that windows automatically executes when a removable drive is inserted thereby infecting the host.

During propagation, Stuxnet 1.x exploited the windows server service RPC vulnerability to establish a P2P network among infected hosts in order to propagate updates to each other. Eventually, Stuxnet 1.x would infect the Step 7 control software in the same manner as Stuxnet 0.5 after being brought via USB devices and inserted into the control system's key hosts in the SCADA network. Then, by modifying the PLC parameter data, Stuxnet 1.x made the centrifuges repeatedly changed rotation frequency between the limit frequency of 1410 Hz, the normal frequency of 1064 Hz, and the low frequency of 2 Hz, which not only resulted in the ineffective separation of U-235 and U-238 from the gas mixture, but also caused the failure rate of centrifuge to skyrocket as a result of the repeated frequency changes.

Stuxnet 0.5 and Stuxnet 1.x worms ultimately achieved the destruction of 1/5 of the centrifuge units and made the separated uranium incapable of meeting weapons-grade requirements. The attacks resulted in the near-permanent stagnation of Iran's nuclear weapons development plan.

Our threat ripple model can characterize existing attack events, clearly show the process of the adversary successfully attacking the platform, then modifying the platform data to make the threat transmit from platforms to businesses and then spreading among the businesses and affecting the architecture. The threat ripple model has the ability to cover both the expressive capability of the Kill Chain model for threat spreading across platforms and the ability of ATT&CK for attacks on platforms, as well as the ability to characterize the ripple processes of threats among businesses and platforms in the architecture. This allows us to accurately characterize attack that aim at creating threat ripple effects in the architecture.

7 Attack inference on representative banking scenarios

The U.S. Department of Homeland Security paper provides a typical modellable attack scenario against large banks and constructs assumptions about the potential systemic effects on the business following an attack on these banks, but lacks a business-oriented model to support it. In this paper, we use it as a representative example of attack prediction to show how, in practice, defender can use our architectural model to model an architecture, characterize the dependencies among businesses in it, and use the threat ripple model to infer an attack on the architecture using the inter-business dependencies.

7.1 Attack scenario

We construct an architecture model of a user-oriented credit card processing business of a large bank for the attack scenario in paper [39], as shown in Figure 8. The model shows the dependencies among the platforms and the businesses. In order to describe it cleanly, we hide the feedback streams of data and code transfer.

The credit card processing business of a large bank is, in fact, a mixture of multiple business dependency sequences that require multiple businesses and platforms to participate collaboratively [40–44]. When a user makes a transaction in an online shopping system, a series of business processes will take place in the banking system:

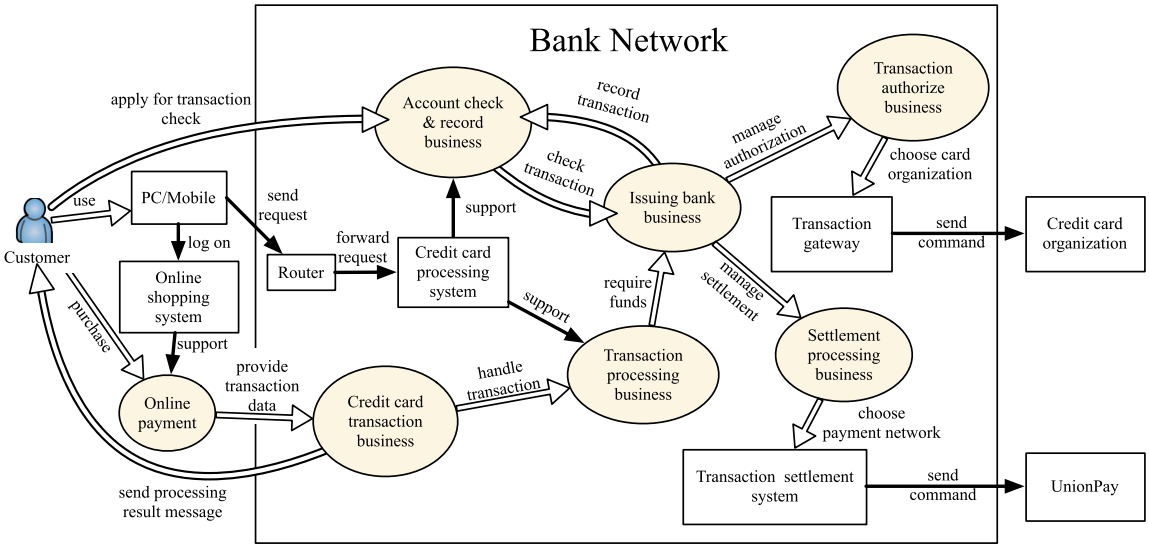

Figure 8. Credit card processing business process for a large bank

The credit card payment will enter the transaction processing stage. First, the transaction data need to be submitted to the bank's credit card transaction business for audit. After the audit is completed, the transaction data will be submitted to the transaction processing business for handling. Next, the issuing bank business will receive the request for funds and after its confirmation it will order the transaction authorization business to select the transaction gateway corresponding to the user's credit card to conduct the electronic transaction with the credit card organization. Then, after the transaction is completed, the account recording business will use the credit card processing system to credit the transaction information to the user's account [42, 43].

At the stage of transaction settlement between the user and the merchant of the online shopping system, the settlement processing business will use the Transaction Settlement System to send an electronic payment request to UnionPay to execute the settlement, after obtaining the approval of the card issuing bank business. When the settlement is completed, the issuing bank business will request the Account Record Business to record the information. All the above processing results will be returned to the user. If there is a dispute over the result of a transaction, the user can apply to the bank for transaction checking and arbitration [40, 44].

These processes follow strict rules. Once an upstream business has problems, it will cause the downstream businesses in its business dependency sequence not being able to operate normally.

7.2 Attack prediction

We combine cases of attacks on financial systems such as the SWIFT event [12, 45–47] and analyses of systemic threats to the banking systems [48–51], make the following attack predictions. All involved techniques in ATT&CK are detailed in Table 1.

The exposed and Attackable Surfaces of the Bank:

- Routers used in bank networks may have vulnerabilities [39].
- Credit card processing system needs to open accessible ports to external networks as it needs to process user requests, which makes it possible for an adversary to access and scan hosts [12, 47].
- A large volume of compromised user credit card information was sold on the black market by criminal groups [45, 46, 50].

The Attackable Surfaces of the online shopping system:

- Vulnerabilities may exist in popular online shopping systems and checkout software [39, 51].

Table 1. Techniques used by adversary in ATT&CK model

ID	Technique Name	Adversary Behaviour
T1590	Gather Victim Network Information	Collect the information of the banking network.
T1595.002	Vulnerability Scanning	Perform vulnerability scan on credit card processing system.
T1592	Gather Victim Host Information	Collect the information of the credit card processing system and online shopping system.
T1597.002	Purchase Technical Data	Purchase technical information about routers deployed in the target bank from the supplier.
T1586	Compromise Accounts	Bring active credit card numbers and consumer identities from the black market.
T1587.004	Exploits	Develop vulnerability exploitation malicious code against the routers, credit card processing system, and the online shopping systems.
T1133	External Remote Services	Leverage external-facing remote services to initially access, exploite the vulnerability to pre-implant malicious code and persist in the routers.
T1190	Exploit Public-Facing Application	Exploit the corresponding vulnerabilities in the credit card processing system and the online shopping system to initially access these hosts and pre-implant malicious code in them.
T1203	Exploitation for Client Execution	Make the online shopping system unable to check the illegal payments.
T1489	Service Stop	Make the routers down and disrupts the network, and paralyze the credit card processing system.
T1490	Inhibit System Recovery	Continually re-insert exploit code into the routers' recovered systems.
T1059	Command and Scripting Interpreter	Trigger the credit card processing system malicious code.
T1078	Valid Accounts	Launch fraudulent on-line transactions using a large number of illegally obtained valid credit card accounts.

The adversary sets the goals of:

- Disrupting credit card business on dates when high volumes of transactions occur.
- Making systemic effects on the bank, and maintaining the disruption over a period of days.
- Undermining bank and consumer confidence.

The adversary preparations:
Information gathering and weaponization, as shown in Figure 9a.

- Purchased technical information about routers deployed in the target bank from the supplier.
- Bought large volumes of active credit card numbers and consumer identities on the black market, complete with credit card validation numbers.
- Developed malicious code that exploits zero-day vulnerability against the version of the system used by the router, and malicious code that targets the credit card processing system and the online shopping system, respectively.

Pre-implantation of malicious code, as shown in Figure 9b.

- Pre-implanted these malicious code into target systems.

7.3 Attack inferring

The adversary's attack strategies and the caused threats ripple within the architecture are as follows.

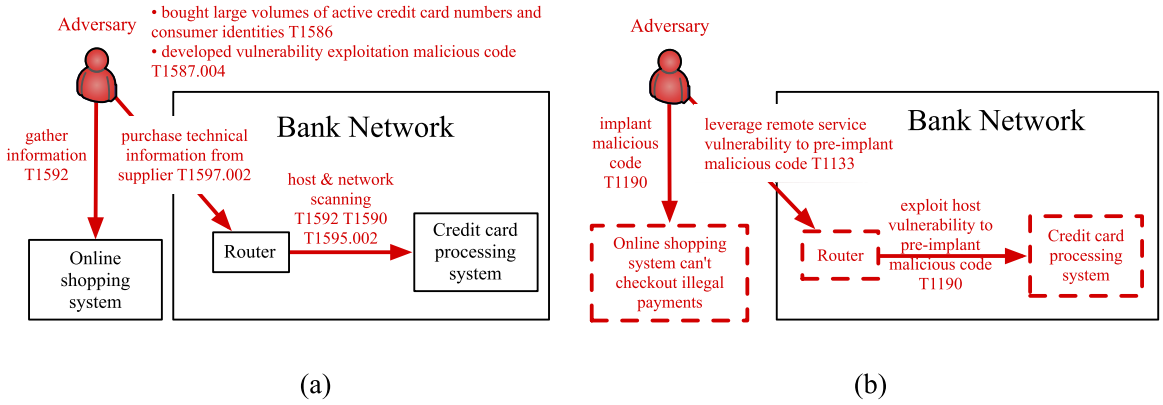

Figure 9. The adversary perparations. (a) Information gathering and weaponization. (b) Pre-implantation of malicious code

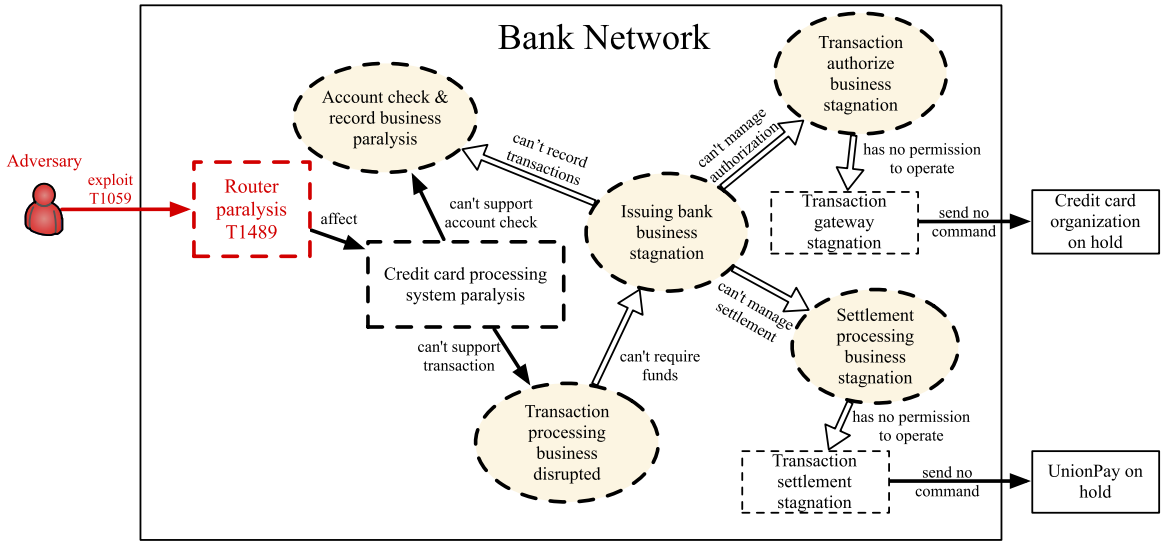

Figure 10. Attack the routers

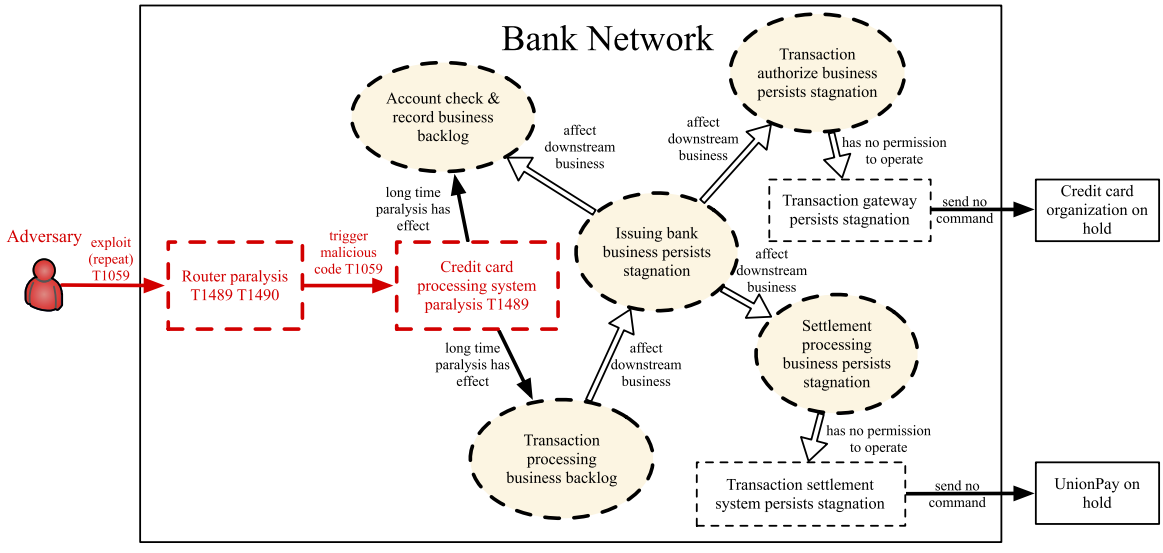

Figure 11. Attack the credit card processing system

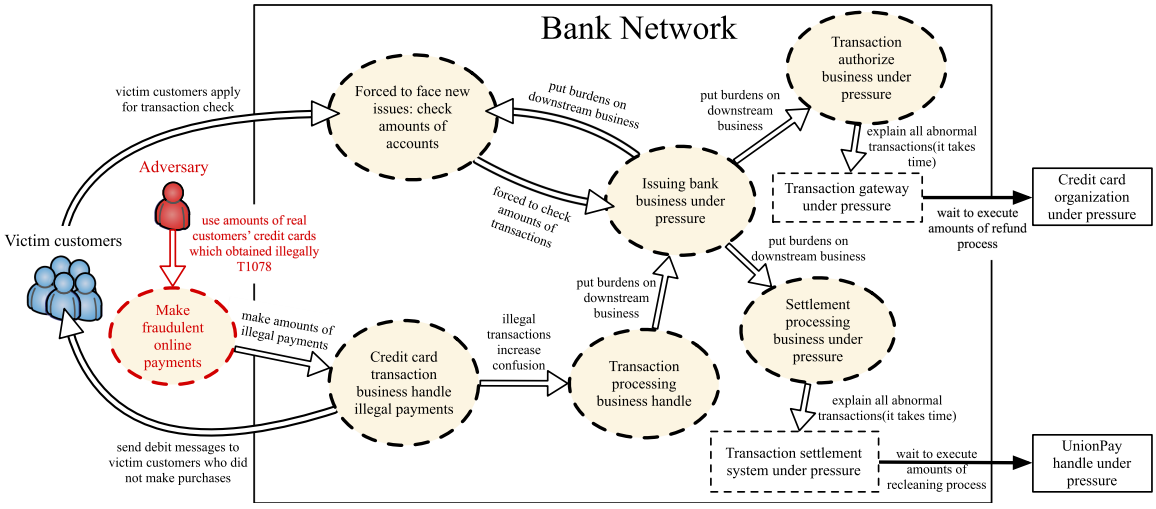

Figure 12. Attack the online shopping system

- **Attack the routers.** On Black Friday, the adversary launches a zero-day attack on the bank's router and brings them down, which disrupts the network and prevents the credit card processing system from operating normally, which stagnates the issuing bank business and downstream businesses that depends on it, as shown in Figure 10. The defender discovers abnormal router running through network diagnostics and brings in backup systems to ensure network recovery as a priority. Meanwhile the adversary is constantly trying to re-insert the exploit into recovered systems of the routers.
- **Attack the credit card processing system.** The adversary triggers the malicious code in the credit card processing system, which paralyses the system and causes a disruption to the transaction processing business it supports, as shown in Figure 11. Along with repeated attacks on routers by the adversary, this makes it difficult for the defender to determine what has gone wrong with the credit card processing system's inability to be accessed by the extranet. The defender works intensely to identify the cause. Once it becomes clear the disruption is due to a cyberattack, the bank takes systems off-line, attempting to find and eradicate the malicious code. The disruption to credit card transactions, due to both the attack itself and the efforts to recover, has not only resulting in a backlog of current business, but also a persistent stagnation of downstream businesses in the business dependency sequence.
- **Attack the online shopping system.** At the same time, the adversary exploits the checkout vulnerability in online shopping system to launch fraudulent on-line transactions using a large number of illegally obtained valid credit card accounts, as shown in Figure 12. Victim consumers enrolled in text alerting services for on-line transactions receive alerts for purchases they did not make and apply to the bank for arbitration of the transactions. This causes the banking system need to perform a large number of transaction verifications under the existing business backlog, which additionally puts burdens on the businesses' processing. The credit card business of the bank cannot return to normal for a short period of time.

The series of attack effects will cause credit card business remain disrupted during major trading days, which will have a serious influence on the bank's financial reputation and consumers' confidence.

We use the ATT&CK model for threat inference on the above attack scenario, as shown in Figure 13. Although it is able to express the transmission of threats between platforms, it only demonstrates the sequence of tactic executions. If combined with the threat ripple model, we can express threats that utilize dependencies among platforms and businesses to have systematic effects on the architecture while showing the attack process. Threat ripple model Integrated all attack phases are presented in Figure 14.

With the construction of architecture model for information system scenarios, coupled with the threat ripple model for attack inference, the defender is able to determine which platforms or businesses in the architecture once attacked will cause threats ripple among the businesses, as well as which factors will result in the persistence of the business affected state. The threat ripple model can help defenders

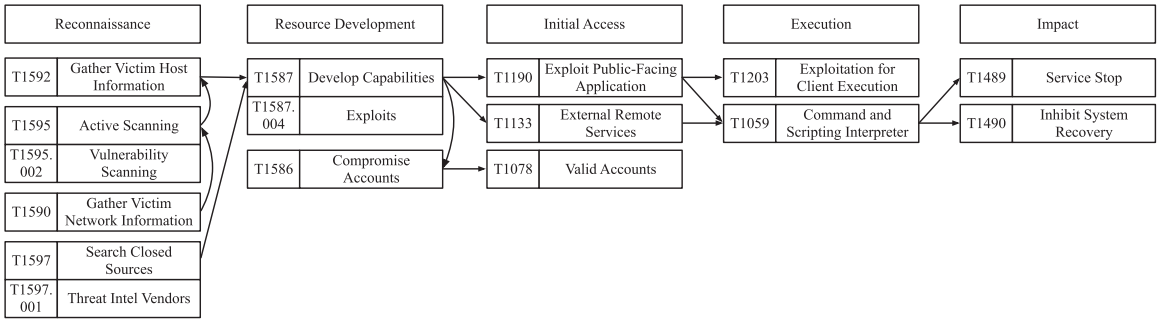

Figure 13. Attack inferring on large bank via ATT&CK model

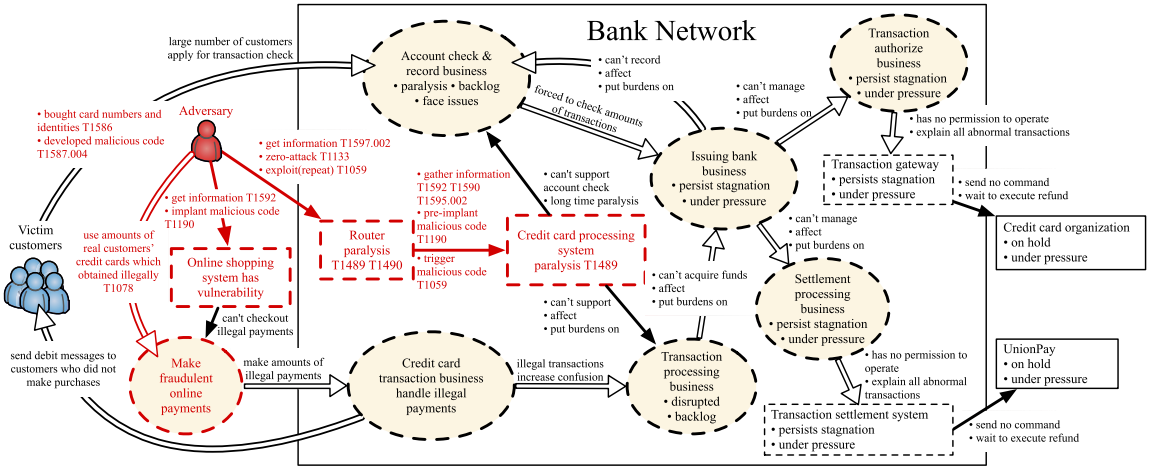

Figure 14. Attack inferring on large bank via threat ripple model

determine the vulnerability of the platform and business, analyze the logic of business interactions within the architecture that might be exploited by an adversary, as well as make strategies to stop the propagation of threats in the architecture.

8 Discussion

In the course of this paper, in order to make the model efficiently applied by professionals, we focus on comparing typical threat models that are widely used in the industry during the design of the model, instead of only focusing on the models and frameworks in the latest papers. Our threat ripple model has applicability to broader and rapidly evolving range of cybersecurity threats by being able to express both the process of infrastructure-oriented cyberattacks and the process of threats using dependencies among platforms and businesses to ripple their effect. To make our modelling approach fully comprehensible to researchers, this paper concisely expresses the logic of threat transmissions among businesses and platforms using function mapping and graphical illustrations, and also shows the application mode of our modelling approach through the modeling characterization of typical attack event and the attack inferring of against important information system:

Attack event analysis:

- Characterize the threat events that have occurred and reviewing the attack process
- Help defenders work out which nodes are the key targets of attackers

System operation and maintenance:

- Determine the dependencies among the businesses and platforms in the target system.

- Discover the transmission paths that enable risks and threats to spread the effects.
- Perform attack inferring on the target system to determine which business or platform once being abnormal will have a ripple effect on other parts of the system, and then strengthen their security.

Respond and assess threat events while encountered:

- Help defenders quantify the business effects of the threats.
- Stop new ripple effects on infrastructure and business in time that might transmit from the threat generated by attack.

In actual scenarios, the use of the model is flexible and researchers can use the system model to decompose the system at the required granularity depending on the complexity of the system, *e.g.*, aggregating businesses that do not need to zoom in detail, thus focusing on vulnerable businesses and platforms.

An unresolved problem is that the business-specific risks and security needs of different industries may vary significantly, which require more tailor-made approaches. Our future work will investigate more domains involving non-information systems. In addition, we will continue work on mathematically expressing the modeling method to make it computable, in order to preparing for introducing algorithms.

9 Conclusion

The threat ripple model can be used to characterize the process of threats transmit and cause effects utilizing the relationships among platforms and businesses. By coupling with the architecture model, it can be used to perform business-oriented attack inference, identify the vulnerability points of the business and platforms in the architecture, and what threat ripple effects will be caused if they are attacked.

For threat events such as Stuxnet and the Ukraine blackout event, the adversary accurately attacked platforms as a means of rippling the threat caused by the attack in the architecture, utilizing the logic of data transmission of the businesses and code transmission of the platforms. Similarly, some attacks activities that aim at cross-modal architectures which are strongly associated with the target architecture, will disrupt its businesses or elements and participants of the business, in order to affect the target architecture, such as assassinating Iranian nuclear weapon scientist causing confusion in the development of nuclear weapons, Türkiye cutting off important water sources in Syria bringing the Kurdish region into a crisis, using social networks to spread false news to affect the public's view on social issues, *etc.* These methods of spreading the effects of an attack via utilizing the relationships among components in cross-modal architectures are essentially threat ripple. Our threat ripple model is suitable for researching these threat ripple processes. In the future, the further exploration in calculation methods of threat ripple effects, important cross-modal business elements, *etc.* are able to form a more dimensional characterization of threat ripple.

Acknowledgments

Thanks to Master Sheng Huang for his support of this research.

Funding

This work was supported by the National Natural Science Foundation of China under Grants No. 62302122, and the Key R&D Program of Heilongjiang Province of China under Grants No. JD2023SJ07.

Conflicts of interest

The authors declare that they have no conflict of interest.

Data availability statement

No data are associated with this article.

Authors' Contributions

Shiliang Ao and Binxing Fang provided the threat modelling methodology. Xinguang Xiao provided the revision and proofreading, and Hongli Zhang provided resource support for this paper.

References

[1] Al Fikri M, Putra FA and Suryanto Y et al. Risk assessment using NIST SP 800-30 revision 1 and ISO 27005 combination technique in profit-based organization: Case study of ZZZ information system application in ABC agency. Procedia Comput Sci 2019; **161**: 1206–15.

[2] Caralli RA, Stevens JF and Young LR et al. Introducing Octave Allegro: Improving the Information Security Risk Assessment Process. MA: Hansom AFB, 2007.
[3] Nagaraju V, Fiondella L and Wandji T. A survey of fault and attack tree modeling and analysis for cyber risk management. In: 2017 IEEE International Symposium on Technologies for Homeland Security (HST), IEEE, 2017: 1-6.
[4] Souppaya M and Scarfone K. Guide to enterprise telework, remote access, and bring your own device (byod) security. NIST Spec Publ 2016; **800**: 46.
[5] Zhang L, Taal A and Cushing R et al. A risk-level assessment system based on the STRIDE/DREAD model for digital data marketplaces. Int J Inf Secur 2022; **21**: 509–525.
[6] Abhishta A, Joosten R and Dragomiretskiy S et al. Impact of successful ddos attacks on a major crypto-currency exchange. In: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2019: 379–84.
[7] Akiyama M, Yagi T and Yada T et al. Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots. Comput Secur 2017; **69**: 155–73.
[8] FireEye. Highly evasive attacker leverages SolarWinds supply chain to compromise multiple global victims with SUNBURST backdoor. FireEye Threat Research, 2020.
[9] Ghafoor I, Jattala I and Durrani S et al. Analysis of OpenSSL Heartbleed vulnerability for embedded systems. In: 17th IEEE International Multi Topic Conference 2014. IEEE, 2014: 314–19.
[10] Gui X, Liu J and Chi M et al. Analysis of malware application based on massive network traffic. China Commun 2016; **13**: 209–21.
[11] Mohurle S and Patil M. A brief study of wannacry threat: Ransomware attack 2017. Int J Adv Res Comput Sci 2017; **8**: 1938–40.
[12] Antiy report, 2019. https://www.antiy.cn/research/notice&report/researchreport/20190601.html.
[13] Falliere N, Murchu LO and Chien E. W32. stuxnet dossier. White paper, symantec corp. Secur Response 2011; **5**: 29.
[14] Langner R. Stuxnet: Dissecting a cyberwarfare weapon. IEEE Secur Privacy 2011; **9**: 49–51.
[15] Liang G, Weller SR and Zhao J et al. The 2015 Ukraine blackout: Implications for false data injection attacks. IEEE Trans Power Syst 2016; **32**: 3317–18.
[16] McDonald G, Murchu LO and Doherty S et al. Stuxnet 0.5: The missing link. Symantec Rep 2013. https://docs.broadcom.com/doc/stuxnet-missing-link-13-en
[17] Yadav T and Rao AM. Technical aspects of cyber kill chain. In: Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3. Springer International Publishing, 2015: 438–52.
[18] Gu G, Porras PA and Yegneswaran V et al. Bothunter: Detecting malware infection through ids-driven dialog correlation. In: USENIX Security Symposium. 2007; **7**: 1–16.
[19] Iskhakov A and Iskhakov S. Data Normalization models in the security event management systems. In: 2020 13th International Conference "Management of large-scale system development" (MLSD). IEEE, 2020: 1–5.
[20] Hutchins EM, Cloppert MJ and Amin RM. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues Inf Warfare Secur Res 2011; **1**: 80.
[21] Assante MJ and Lee RM. The industrial control system cyber kill chain. SANS Inst InfoSec Reading Room 2015; **1**: 2.
[22] Pols P and van den Berg J. The unified kill chain. CSA Thesis, Hague, 2017: 1–104.
[23] Strom BE, Applebaum A and Miller DP et al. Mitre att&ck: Design and philosophy. In: Technical report. The MITRE Corporation, 2018.
[24] Kotenko I and Doynikova E. The CAPEC based generator of attack scenarios for network security evaluation. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). IEEE, 2015; **1**: 436–41.
[25] Wynn J, Whitmore J and Upton G et al. Threat assessment and remediation analysis (tara). MITRE Corporation, 2014.
[26] Bodeau DJ, McCollum CD and Fox DB. Cyber threat modeling: Survey, assessment, and representative framework. Mitre Corp, Mclean, 2018: 2021-11.
[27] Bodeau D and Graubart R. Cyber Prep 2.0: Motivating Organizational Cyber Strategies in Terms of Preparedness. MITRE CORP BEDFORD MA, 2017; 15-0797.
[28] Wichers D. Owasp top-10 2013. OWASP Foundation, February, 2013: 12.
[29] Shevchenko N, Chick TA and O'Riordan P, et al. Threat Modeling: A Summary of Available Methods. Software Engineering Institute— Carnegie Mellon University, 2018: 1–24.
[30] LeMay E, Ford MD and Keefe K et al. Model-based security metrics using adversary view security evaluation (advise). In: 2011 Eighth International Conference on Quantitative Evaluation of SysTems. IEEE, 2011: 191–200.
[31] Stix: Assets affected in an incident. 2018. http://stixproject.github.io/documentation/idioms/affected-assets/.

[32] Kotusev S. TOGAF-based enterprise architecture practice: An exploratory case study. Commun Assoc Inf Syst 2018; **43**: 20.
[33] Tao ZG, Luo YF and Chen CX et al. Enterprise application architecture development based on DoDAF and TOGAF. Enterprise Inf Syst 2017; **11**: 627–51.
[34] Veronica AI and Ugochukwu O. Design and Development of a Web-Based Information System for Security Agencies. Technical & Industrial Sponsors, 2016: 237.
[35] Haes Alhelou H, Hamedani-Golshan ME and Njenda TC et al. A survey on power system blackout and cascading events: Research motivations and challenges. Energies, 2019; **12**: 682.
[36] AlMasri TN and AlDalaien MN. Detecting Spyware in Android Devices Using Random Forest. In: International Conference on Advances in Computing Research. Cham: Springer Nature Switzerland, 2023: 294–315.
[37] Alkhadra R, Abuzaid J and AlShammari M et al. Solar winds hack: In-depth analysis and countermeasures. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2021: 1–7.
[38] Xiao C. Malware xcodeghost infects 39 ios apps, including wechat, affecting hundreds of millions of users. PaloAlto Network Unit, 2015; 42.
[39] Bodeau DJ and McCollum CD. System-of-systems threat model. The Homeland Security Systems Engineering and Development Institute (HSSEDI) MITRE: Bedford, MA, USA, 2018.
[40] Brauchle JP, Gbel M and Seiler J et al. Cyber mapping the financial system. Carnegie Endowment Int Peace, 2020. https://carnegie-production-assets.s3.amazonaws.com/static/files/Brauchle_Cyber_Mapping_the_Financial_System_final.pdf
[41] Gulyas O and Kiss G. Impact of cyber-attacks on the financial institutions. Procedia Comput Sci 2023; **219**: 84–90.
[42] Loader D. Clearing, Settlement and Custody. Butterworth-Heinemann, 2019.
[43] Priem R. Distributed ledger technology for securities clearing and settlement: benefits, risks, and regulatory implications. Financ Innovation, 2020; **6**: 1–25.
[44] Wewege L, Lee J and Thomsett MC. Disruptions and digital banking trends. J Appl Finance Banking, 2020; **10**: 15–56.
[45] Desai N. Understanding the theoretical underpinnings of corporate fraud. Vikalpa, 2020; **45**: 25–31.
[46] Hashim HA, Salleh Z and Shuhaimi I et al. The risk of financial fraud: a management perspective. J Financ Crime, 2020; **27**: 1143–59.
[47] Kellermann T and Murphy R. Modern bank heists 3.0. Annual "Modern Bank Heists". VMware Carbon Black, 2020.
[48] Ghelani D, Hua TK and Koduru SKR. Cyber security threats, vulnerabilities, and security solutions models in banking. Authorea Preprints, 2022. https://doi.org/10.22541/au.166385206.63311335/v1
[49] Melnyk LV, Gudz IF and Synchak VP et al. Financial security and guidelines of the strategic development of Ukraine. Rev Econ Finance 2022; **20**: 1264–77.
[50] Pomerleau PL and Lowery DL. Countering cyber threats to financial institutions. In: A Private and Public Partnership Approach to Critical Infrastructure Protection. Springer, 2020.
[51] Shkolnyk IO, Kozmenko SM and Polach J et al. State financial security: Comprehensive analysis of its impact factors. 2020; **13**: 291–309.

Shiliang Ao received the B.S. degree from Xidian University, Xi'an, China, in 2015. He is currently a Ph.D. candidate in the School of Cyberspace Science from Harbin Institute of Technology (HlT). His research interests include cybersecurity and system engineering.

Binxing Fang received the Ph.D. degree in computer science and technology from Harbin Institute of Technology, Harbin, China, in 1989. He is a member of the Chinese Academy of Engineering. His cument research interests include computer networks, information and network security.

Xinguang Xiao graduated from Harbin Institute of Technology, Harbin, China, in 1998. He is an adjunct professor in HlT, and the Chief Architect of Antiy Technology Group Co., Ltd. His research interests include Advanced Persistent Threat (APT) analysis & traceability and cyberspace security engineering.

Hongli Zhang received the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1999. She is currently a Professor with the School of Cyberspace Science in HlT. Her research interests include network and information security, network measurement and modeling.