Cyber Weapons as Components of Weapon Systems

Fabrizio Baiardi, Vincenzo Sammartino

Dipartimento di Informatica, Università di Pisa, Pisa, Italy
f.baiardi@unipi.it
vincenzo.sammartino@phd.unipi.it

Research Track: CSCI-RTCW: Cyber Warfare, Cyber Defense, & Cyber Security

Abstract. Viewing a cyber weapon as a component of a weapon system mitigates some of the potential weaknesses of these weapons, such as their fragility or manipulations to circumvent their functions. This is confirmed by experience with malware strains and their evolutions, where the attack infrastructure of the actor using the weapon mitigates vulnerabilities in these strains. This is possible because this infrastructure enables the attacker to coordinate and update the various modules of the weapon. This is possible even after deploying the weapon on target systems or when it spreads in the internet to reach its targets. Adopting this infrastructure is aligned with other physical weapons that are also supported by an infrastructure. We also discuss the advantages enabled by the one-to-many nature of a cyber weapon and the modules that define this weapon.

Keywords: Cyber weapon · Payload · Attack infrastructure · Spread · Deployment · One-to-many weapon.

1 Introduction

Several researchers have investigated the topic of cyber weapons and discussed their effectiveness. This effectiveness is crucial because these weapons exploit specific vulnerabilities to spread in computer networks and persist in target systems. Still a weapon becomes ineffectual once the owners of the target systems address the vulnerabilities it exploits. This is the reason why some researchers describe a cyber weapon as single-use because its usage reveals the vulnerabilities it exploits, which leads to their remediation. Similar reasoning explains someone defines cyber weapons as fragile because a new release of the target systems may make them ineffective [1,10,17,21,24,23,31,34,38]. These conclusions focus too much on a single component of a cyber weapon, namely its payload. Distinct conclusions are reached if we see the payload as just one module of a weapon system with other components, such as a diffusion vector and the attack infrastructure. This infrastructure enables interactions with a weapon [12,30] to update the payload and other modules to preserve effectiveness despite the

remediation of the vulnerabilities the current versions of the weapon modules exploit. Furthermore, the time to produce and apply a patch, the most common remediation, is measured in days, if not weeks, while new versions of a weapon may be generated quickly and employed against various targets within a brief timeframe, rendering comprehensive mitigation for all the targets impractical. Lastly, even a single use cyber weapon may hit a large, or even a huge, number of targets by implementing a one-to-many attack [2].

Fig. 1. Glossary of Abbreviations

Abbreviation	Description
cw	a cyber weapon
Ij(cw)	one instance, i.e. a copy of cw
A(cw)	the actor, i.e. a state, that has produced cw
At(A(cw))	the attack infrastructure to interact with the instances of cw
AutCt(cw)	the module that discovers a node is a target
C2(cw)	the module to interact with the infrastructure and download new versions
Di(cw)	the module to spread cw in a network to reach its targets
Pay(cw)	the payload module of cw that is executed to produce the impact
Pe(cw)	the module that escalates the privileges to install modules on a target
Sp(cw)	the module to spread instances of cw in a network to reach its targets
T(cw)	the targets of cw, where to install and execute the payload
tde(cw)	the starting time of the design of cw
tdp(cw)	the time when cw is available, i.e. it has been designed and implemented
tf(cw)	the time when the execution of Pay(cw) can be fired

From our perspective, a cyber weapon is a complex, modular system. The main element of this system is an attack infrastructure. This is a botnet that connects network nodes the attacker controls and uses to stealthily interact with the cyber weapons it has deployed. Sophisticated attack infrastructures integrate secure communication channels, fail-safe protocols, and evasion methods to maintain continuous access to certain nodes while avoiding detection by security monitoring systems [12,14,16,19,29]. The attack infrastructure supports the dissemination of the weapon's instances, each including a payload. Instances spread across network nodes until some instances successfully infect and persist on some nodes, the targets of the weapon. The attacker then uses the infrastructure to interact with these instances to choose if and when to activate the payload and produce effects, such as collecting and exfiltrating data, destroying systems, and overwriting or corrupting data. The overall impact depends on whether the targets are cyber or cyber-physical systems [36]. The view of a cyber weapon as part of a weapon system aligns with the knowledge we have gained from the different strains of malware, their development, and updates. This confirms the critical role of the attack infrastructure in the production and deployment of variants of malware.

This paper proceeds as follows. Sect. 2 discusses the adoption of an attack infrastructure, while Sect. 3 outlines the modular structure of a cyber weapon and the role of each modules. Then, Sect.4 describes the design space of a weapon,

while Sect. 5 considers some attributes of a cyber weapon. Sect. 6 investigates alternative compromises among contrasting requirements. Figure 1 resumes the abbreviations we use.

1.1 Our contribution

The primary contribution of this paper lies in the examination of matters pertinent to the creation and dissemination of a cyber weapon, by treating it as a modular component within a weapons system. We show that the effectiveness and fragility of cyber weapons can be addressed if the weapon payload is just one module of a system built around an attack infrastructure. This infrastructure can replace and improve this module, as well as other ones, after its deployment on targets and before its execution.

2 The Attack Infrastructure of a Cyber Weapon

This section discusses the role of the attack infrastructure in a general scenario, where a lethal autonomous weapon system (laws) [22,30] that is introduced into a network, usually the Internet, and navigates through it to reach its targets. We consider strategic cyber weapons with an impact far from the battlefield against the critical infrastructures of a nation. In the following, we assume that attack strategies of cybercrime gangs can also be applied to a cyber weapon. This assumption is based on the tight relations between these gangs and state-sponsored operators [9,26]. Several pieces of evidence support this assumption. As an example, several countries have most frequently drawn on resources from criminal forums.

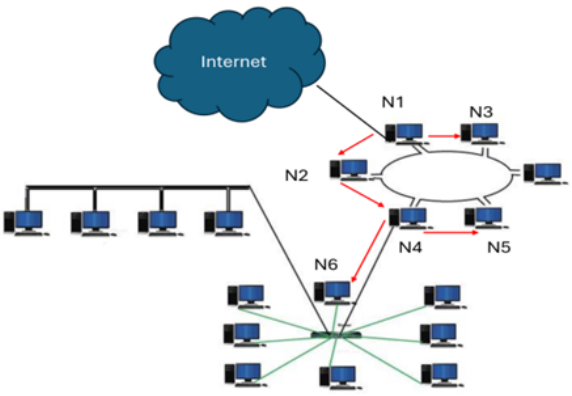

Fig. 2. Spreading of a cyber weapon in a network.

2.1 A weapon and its attack infrastructure

A cyber weapon cw is a system that an adversary A(cw) builds to manipulate some ICT/OT systems, T(cw), the targets of cw. The number and the kinds of systems in T(cw) determine the impact of cw. Anyway, the existence of several targets stresses the one-to-many nature of cyber weapons that usually hit a set of targets in a coordinated manner. The impact of cw may be limited to the cyber world, i.e. a wiper that erases documents and databases, or it may also affect the physical world, i.e. a malware to manipulate an industrial control system to destroy it [36]. The size of T(cw) may be initially unknown and be determined according to attributes of the network nodes that cw crosses when spreading, i.e. according to the applications they run.

In general, at any moment after starting the spreading of cw by deploying it on a network node, several instances, i.e. copies, of cw spread in the network to reach their targets. An instance of cw running on a node N1 moves to node N2 by executing a remote attack from N1 to N2. If the attack is successful, the instance on N1 creates a copy of itself on N2. An instance attacks several nodes each time there is a fork in the paths it can follow, and it creates a new instance on each node it has successfully attacked. This implies that at any moment, n instances I1(cw), ..., In(cw) of cw may run in parallel on distinct nodes. A(cw) can set or not an upper bound for n a priori. Usually, no bound is set when A(cw) cannot access accurate information on T(cw) or on the paths to reach these targets. Anyway, to minimise both effort and noise, A(cw) can insert some controls in cw to avoid repeating attacks against a node it has already reached. If the controls are effective, each node executes at most one instance of cw. In the example in Figure 2, an instance in N1 creates two instances after attacking N2 and N3, while the instance in N4 creates instances on N5 and N6.

We assume that A(cw) has built and uses an attack infrastructure At(A(cw)), see Figure 3, to interact with the instances I1(cw), ..., In(cw). The adoption of At(A(cw)) transforms cw into an element of a weapon system. At(A(cw)) usually consists of a botnet of nodes with distinct owners. A(cw) controls these nodes it

Fig. 3. An attack infrastructure (Bot nodes) that Interfaces A(cw) (Botmaster) with the nodes where cw has been installed or with those it crosses in its spreading (Victim)

has previously attacked to install tools that support stealth interactions between A(w) and I1(cw), ..., In(cw) to exchange information, update their modules and coordinate their operations. At(A(cw)) hides A(cw) behind nodes controlled by other organisations to obfuscate, e.g. increase the complexity of, attribution of cw. However, At(A(cw)) may exist even when obfuscation is useless, as it speeds up the interactions between A(cw) and I1(cw), ..., In(cw) [6]. Distinct weapons built by A(cw) usually share, at least partially, the same attack infrastructure. The diamond model of cyber intrusions [4] confirms the role of At(A(cw)) as one of the four core features of any cyber intrusion: adversary, infrastructure, capability, and victim. Most of the infrastructures that threat actors use to interact with malware or ransomware they deploy include legitimate Internet services such as Telegram, GitHub, or OneDrive [13,16,19,20,25,29,35]. This blends the traffic of At(A(cw)) with the normal one as a first countermeasure against the discovery and seizure of domains and servers in At(A(cw)). Command and control communication in these legitimate services is seldom advanced and rarely sophisticated, but it strategy adds a layer of complexity for defenders as it increases the risk of misidentifying malicious activity as legitimate.

3 Modules of a cyber weapon

We discuss a modular decomposition of a weapon and point out how the existence of an attack infrastructure can support and simplify these modules. Simpler weapons will be discussed as well.

We extend [11] by functionally decomposing cw into four modules:

1. Command and control C2(cw).
2. Spreading Sp(w).
3. Persistence Pe(cw).
4. Payload Pay(cw).

Each module can be further decomposed into simpler ones to increase flexibility and software reuse. Our decomposition also supports the definition of the four important features of a cyber weapon, which are:

a) The role of At(A(cw)).
b) How cw spreads.
c) How cw persists in its target.
d) The payload of cw.

In the following, we show how At(A(cw)) can tune and update the last features.

Command & control module C2(cw) interacts with A(cw) through At(A(cw)) to exchange information and to download and install updated versions for any module of cw. C2(cw) communicates to A(cw) information on any node where an instance is running as soon as the instance is created. In general, the communications are encrypted to preserve integrity and confidentiality.

Spreading module Sp(cw) has to spread cw across the Internet. It can be further decomposed into two modules:

1. Diffusion module Di(cw),
2. Autonomous control module AutCt(cw).

Di(cw) attacks another node to generate another instance of cw on the target. It typically exploits one or several vulnerabilities that grant remote code execution on the target. They are classified as wormable, referring to malware that replicates itself on other nodes. Any new instance of cw includes all its modules. To avoid an uncontrolled spreading, Di(cw) may even destroy, ie kill, an instance that has not yet reached a target at the end of a time interval . This event may also be fired by commands from A(cw) through At(A(cw)).

When Di(cw) creates a new instance of cw on a node, it activates AutCt(cw). AutCt(cw) decides if the underlying node belongs to T(cw) and, if so, it activates Pe(cw). Then, it activates Di(cw) to spread cw to other nodes. The simplest version of AutCt(cw) collects data on the underlying node and transmits it to At(A(cw)), where the decision is taken. Other versions may use more sophisticated or even AI strategies [28,31,32], but simpler strategies to discover targets have been successfully used for a long time.

In the spreading in Figure 2, an instance of cw is forked on N1 through an Internet connection. This instance attacks both N2 and N3 and creates one instance on each of these nodes. Both instances spread autonomously. As an example, if all the remote attacks are successful, the instance running on N2 creates two instances on N4 and N6 while the one on N4 creates two instances on, respectively, N5 and N5. If one instance in Figure 2 runs on a node with an Internet connection, it can use it to spread to other networks.

Di(cw) and AutCt(cw) cooperate because Di(cw) tries to spread instances of cw on any network node, while AutCt(cw) prunes instances on nodes that do not belong to T(cw). cw uses these nodes just as stepping stones to its targets. Further mechanisms to prevent an out-of-control spread of cw are:

1. Each instance is paired with a counter and spreads only if it is positive. Any time it creates another instance, it decreases the counter and pairs with the new instance.
2. Di(cw) never attacks nodes with an IP address in a predefined set.
3. AutC(cw) never activates the persistence module on nodes with some features, i.e. on nodes with a keyboard with a given alphabet.

Persistence module Pe(cw) installs C2(cw), AutC(cw), and Di(cw) on each node that cw crosses as it spreads. Instead, it installs Pay(cw) only if AutC(cw) recognizes the node as a target. If this requires some further privileges, Pe(cw) implements the corresponding escalation. Some weapons do not include Pe(cw) because AutC(cw) immediately fires the execution of Pay(cw). New versions of Pe(cw) that A(cw) downloads through At(A(cw)) may exploit new vulnerabilities. In this way, cw take advantage of the time between the discovery and the patching of a vulnerability.

Payload module Either Pe(cw) immediately runs Pay(cw) after implementing the proper attack chain, or it stealthily installs both Pay(cw) and C2(cw). After being installed, Pay(cw) can interact through At(A(cw)) with A(cw). This enables the update of Pay(cw) to enrich the vulnerabilities it can exploit as well as the synchronisation of the execution of payloads on distinct targets. This is related to the one-to-many feature of cw, and it may be fundamental to achieving the overall impact of interest. Maximum flexibility is achieved when Pay(cw) acts as a beacon that transmits information on the target node to A(cw) that then selects the best payload to install.

Dependencies The modules of cw are largely independent, as new versions of each one can be developed and implemented separately. Furthermore, the vulnerabilities that Sp(cw) exploits to spread cw are independent of those in Pay(cw) to produce an impact. This implies that some modules may have already been developed or that a module can be developed in parallel with the payload. This is a fundamental advantage of the usage of At(A(cw)).

3.1 Features of a cyber weapon

Our decomposition is coherent with the defining characteristics of cyber weapons [38] as well as with the cyber kill chain [15] that defines, see Figure 4, seven stages: Reconnaissance, Weaponisation, Delivery, Exploitation, Installation, Command and Control (C2), and Actions on Objectives. According to [36], the four features of a cyber weapon are:

1. They may combine multiple payloads for espionage, data theft, or sabotage.
2. A stealth capability to persist within the targets over an extended period.
3. An detailed knowledge of the workings of the targets.
4. Proper code to bypass security controls.

Distinct payloads result from merging various functions in Pay(cw) and then coordinating them via At(A(cw)). The stealth capability of spreading and achieving persistence is delegated to Di(cw) and Pe(cw). The development of each module requires data on how the targets operate.

4 The Design Space of Cyber Weapons

If At(A(cw)) is already available, three are the key times in the life cycle of cw.

a) tde(cw), when the design of cw starts.
b) tdp(cw), when cw may be deployed, i.e. it has been designed and implemented, and its spreading can start.
c) tf(cw), when the execution of Pay(cw) can be fired because cw is installed on all its targets.

In the following, we discuss how At(A(cw)) impacts these times. We assume cw is spread as soon as it is available. However, there are good reasons to delay the spreading, because this strongly reduces the probability that the adversary detects the installation of cw.

Fig. 4. Cyber Kill Chain [32]

4.1 The simplest weapons

The most basic cyber weapons are simple payloads which are physically delivered to a target system — commonly via portable devices like USB keys—and executed immediately. A more complex scenario arises when an attack chain is needed to acquire the privileges to execute Pay(cw). These weapons encompass both Pe(cw) and Pay(cw). They may also include C2(cw) to coordinate the payload executions. More advanced cyber weapons utilize social engineering tactics, such as phishing or spear-phishing, to deliverPay (cw). They exploit human vulnerabilities to facilitate initial access, making them a prevalent approach in targeted attacks.

4.2 Spreading through intrusions

The next level in complexity is a weapon that spreads not in an automated way but through intrusions by human attackers. This is how the LockerGaga malware has been deployed in the intrusion against Norsk Hydro in March 2019 [33,36]. Each successful intrusion installs Pay(cw) or both Pay(cw) and C2(cw) in one target in T(cw). The time to deploy these weapons is one to implement all the intrusions. This strategy can accelerate malware dissemination, especially when T(cw) comprises a limited number of nodes, as it allows for the installation of Pay(cw) and C2(cw) to synchronize their activities. Actors targeting critical infrastructure often adopt this method to establish persistent access and preposition capabilities for future geopolitical contingencies.

4.3 Supply chain spreading

Other weapons that only include Pay(cw) and C2(cw) are those deployed through a supply chain attack [8]. This attack injects the two modules within one module of the target system during the system development and before its deployment. This approach involves injecting Pay(cw) into the system's supply chain, often affecting a large number of nodes — far exceeding the size of T(cw). The role of C2(cw) in this context is to interact with At(A(cw)) to determine whether

Pay(cw) has been installed on a target and when to activate it. As an example, a backdoor planted through a supply chain attack in Magento plugins has been dormant for six years before being exploited to attack 11.000 online stores [5]. Other weapons install both Pay(cw) and AutCt(cw). The latter determines autonomously if it has been installed on a node in T(cw).

4.4 Remediating an air gap

The targets of some weapons are not connected to the Internet. Usually, in these weapons, Sp(w) creates an instance of cw and stores it in any mobile memory unit when it is connected to a node that runs the instance. A typical memory unit is a device of an outsourcer working for several companies and that, at distinct times, it is connected to several networks. This device can support the spreading of cw to any customer network it is connected to, even if the network is not connected to the Internet. This strategy is used anytime there is an air gap between a target node and the Internet. These weapons do not include C2(cw). In some weapons, C2(cw) exists, and it copies both data and an instance of cw to mobile memories. This includes memories such as a USB key or even the one on any smartphone or a pc that is connected to the node. Anytime the same memory is connected to another node, the new instance will be transferred to this node. If and when the memory is connected to a node that has an Internet connection, the instance is installed in the node and then uses the connection to transmit the data in the memory to At(A(cw)). This can create a bidirectional channel even to/from nodes in T(cw) protected by an air gap.

4.5 Late payload download

This approach spreads a cyber weapon without Pay(cw), which is downloaded and installed later through C2(cw). This method allows for overlapping development and dissemination phases, enabling Pay(cw) to be tailored to specific targets. It also introduces the concept of weapon-as-a-service, where cyber mercenaries provide customizable attack capabilities on demand.

4.6 Mass weapons

Some weapons implement a mass attack, i.e. they attack a node where they spread. As an example, some wipers aim to erase information from any node where they are installed. Now cw misses AutCt(cw) and it installs Pay(cw) on each node where it spreads.

5 Attributes of Cyber Weapons

The effectiveness, cost, and overall impact of cyber weapons are primarily determined by their attributes, which are influenced by the capabilities of At(A(cw)). These attributes include power, impact, fragility, latency, development time, control, autonomy, robustness, and the role of attack infrastructure.

5.1 Power and impact

The power, or intensity [37], of cw is evaluated by the number of its targets where it has installed Pay(cw). The ratio between this number and the size of T(cw) is a simple assessment of the power at a given time and it depends upon the number of vulnerabilities that Di(cw) and Pe(cw) can exploit, respectively, to spread cw and install Pay(cw) on a target node. A(cw) can measure the current power of cw if C2(cw) is installed not only on the nodes in T(cw) but also on any node where cw spreads and informs A(cw) about the deployment. If A(cw) believes that the current power is too low, it can download new versions of Di(cw) and/or Pe(cw). This flexibility is paid for by interactions through Att(A(cw)) that increase the probability of detecting the spreading. We distinguish the power of cw from its impact because this depends mainly upon the kind of systems in T(cw). As an example, if Pay(cw) manipulates some databases that drive the behaviour of an industrial control system, we can expect a larger impact than when the databases are those of an e-commerce site. This is true even when the same payload is used so that the same code can be considered a weapon or a simple nuisance depending on the target.

5.2 Fragility

Any assessment of fragility of cw should consider that the adoption of both At(A(cw)) and C2(w) enable updates of Sp(cw), Pe(cw) and Pay(cw). In turn, this implies that A(cw) may produce and spread new versions of these modules that use new exploits to take advantage of distinct vulnerabilities. At(A(cw)) enables A(cw) to update any module of cw to remove bugs or increase the number of vulnerabilities it exploits. Furthermore, A(cw) can choose new targets for some or all the instances. The ability to produce and spread new versions of cw is due not only to At(A(cw)) but also to the flexibility of the modules of cw that, in the end, are code fragments. This enables A(cw) to update both T(cw) and the modules of cw. Hence, while each version of cw is fragile, the overall weapon may be highly robust due to updates through At(A(cw)). This stresses the importance of At(A(cw)) and the view of cw as just one module of a weapon system. It also shows that the fragility of cyber weapons is strongly influenced not only by the capability of the defenders but also by how weapons are deployed.

The critical role of the infrastructure is confirmed by available data on the most effective malware. As an example, Emotet first appeared in 2014 as a banking trojan, and it has evolved into one of the largest malware-as-a-service infrastructures [3,37]. The threat actors behind Emotet implemented a series of attack waves that delivered a variety of different payloads, including IcedID, TrickBot, UmbreCrypt and QakBot. There were periods of inactivity interspersed within the attacks to be both stealthy and persistent. Emotet was active until early 2021. In January 2021, law enforcement targeted its attack infrastructure in a coordinated takedown effort, Operation Ladybird. The malware had a resurrection in 2022 before its definitive end. Three main variants of Emotet and several

minor ones have been deployed that used more than twenty attack techniques [27]. The long life and prolonged persistence of successive malware versions confirm the potential robustness of a cyber weapon because Emotet may be seen as a weapon that has been spread eight years before its activation, properly maintained and updated. This exemplifies how an infrastructure can assure the effectiveness of a weapon.

5.3 Latency

This is the time to reach all the elements in T(cw) and it is the difference between tf(cw) and tde(cw). Latency affects the effectiveness of cw if the size and the elements in T(cw) are decided in advance, and it strongly depends upon the number of vulnerabilities that Sp(cw) can exploit. To minimise latency, these vulnerabilities should be highly heterogeneous and affect distinct operating systems and applications [28] as this enables cw to spread through interconnected networks and to exploit any available path to reach a target. Latency is most critical when the execution of Pay(cw) is synchronised in all the targets because it is given by the longest time to reach a target. Anyway, the time to patch all the systems to prevent the spreading of cw is usually several orders of magnitude larger than the latency provided that all the targets can be reached through the network and the vulnerabilities that Sp(cw) can exploit enable cw to reach its targets.

5.4 Time to develop

This time is given by tdp(cw)-tde(cw). In principle, Pay(Cw) and AutCt(w) are the only modules to be developed to build a new weapon. Even Pe(cw) is developed if cw is the first weapon of A(cw) that exploits the corresponding attack chain. We believe that this is a minor case due to the large homogeneity in ICT networks and nodes. Any other cw module from a previous weapon can be reused. The time to develop Pay(Cw) is related to the power of cw and to the number of vulnerabilities to exploit. If A(cw) has a bag of exploits available, the development time is determined by the number of exploits that A(cw) can use. The time strongly increases if no exploit is available or if new vulnerabilities are needed to spread or install the payload. The time to develop AutCt(cw) mostly depends on the one to define a "signature" of T(cw) namely a compact representation of the features of all and only the nodes in T(cw). This definition may involve new intrusions to collect data or the exfiltration of data from previous intrusions. The attack infrastructure strongly reduces the time to collect this data. Furthermore, latency can be reduced by spreading cw from distinct nodes of At(A(cw)) in parallel. We have already discussed how to reduce the time to develop by spreading a preliminary version of cw where Pay(cw) is missing or it is a simple loader that will download and install the proper version of Pay(cw).

5.5 Control and autonomy

The first attribute assesses the control of A(cw) on the spreading of cw and the triggering of Pay(cw). It may be approximated as the probability that only the nodes in T(cw) are influenced by the execution of Pay(cw) when and if it is fired. The autonomy of cw depends both upon its ability to spread to nodes in T(cw) and to identify nodes in T(cw) without any interaction with A(cw). Full autonomy requires the gathering of a large amount of data to develop and debug the two modules in Sp(cw) because cw cannot interact through At(cw). Instead, if both AutCt(cw) and C2(cw) are installed, AutCt(cw) can, at run time, gather data on the node where it is running and send it to At(cw), which checks if the node is a target. This overlaps the spreading of cw with the collection of information to discover nodes in T(cw) and their signature. Further controls in AutC(cw) can prevent spreading to some IP addresses and some geographical regions.

5.6 Robustness

This attribute assesses the complexity for the owner of the nodes in T(cw) of subverting the behaviour of cw to utilise it against A(cw). Since cw is highly complex, there will be some vulnerabilities in some of its modules. This may lead to the discovery of the spreading of cw, the capture of some instances and their reverse engineering to convert cw into a weapon to be launched against those who created it. Strategies to slow down the reverse engineering of cw include obfuscation and polymorphism of the code of cw [6]. At(A(cw)) affects this attribute too because the time to subvert the behaviour of cw mainly depends upon the spreading time, and the attack infrastructure can reduce this time.

6 Design Choices and Attack Infrastructure

Currently, the primary decision in the design of a cyber weapon concerns whether to use autonomous spreading or intrusions from nodes of At(A(cw)). Autonomous spreading will become a viable option in the future, but currently, a set of intrusions is the preferred approach, at least to speed up the spreading, and each intrusion may be automated through an attack platform. Autonomous spreading may require some compromises. As an example, tf(cw), the time to install cw on its targets is the sum of tdp(cw), when the spreading starts, and the latency. In turn, tdp(cw) is determined by the largest time between the one to develop and the one to define the signature of T(cw) and it can be reduced by approximating the signature to collect less data. This compromise is one of the reasons for the trilemma [21], but multiple factors determine the intensity of cw, including its fragility, rather than the latency and the time for its development only. In any case, the attack infrastructure simplifies and speed up the spreading and the deployment. An accurate assessment of cyber weapons cannot neglect that A(cw) does not dynamically produce a weapon when it needs it in the same way it does

not produce a bomb or a bombardier after selecting the target of the bombing. Instead, A(cw) produces cw in advance together with At(A(cw)). This implies that:

a) Distinct weapons can reuse modules to reduce cost and speed up development.
b) Distinct weapons share the same attack infrastructure.
c) The attributes of a weapon mostly depend on the ability to design and deploy it well in advance of its activation and to update its modules.

Hence, the *just pull the trigger* assumption, i.e. you just need to decide when to use cw, is false because At(cw) has to design, build, and test cw before using it.

We believe the main reason of overlooking the role of the attack infrastructure is Stuxnet [7,18]. This was one of the first cyber weapons that became public and has been extensively examined, as the first known example of malicious code designed to leap from some computers to the physical world to cause physical impact on the industrial equipment those computers controlled — centrifuges — to have a kinetic impact on them. Stuxnet stands out as an unusual weapon because its intended target was offline. Hence, several conclusions drawn from Stuxnet do not accurately represent the scenario defined by malware evolution. Malware strains, i.e. variants, are a more realistic example when analysing the strengths and weaknesses of cyber weapons and the role of an attack infrastructure to increase the effectiveness of cw by providing additional computing power and more precise information about targets as well as alternative versions of modules to download. The critical role of infrastructures is also confirmed by the continuous evolution of techniques to build them and to minimise the probability that they are detected. As an example, the attack infrastructures of several APT groups are operational relay boxes, orb, networks [19,25]. These are botnets that consist of virtual private servers together with compromised Internet of Things devices, smart devices, and routers. Most of these nodes are end-of-life or unsupported by their manufacturers. In this way, the attackers can grow the size of the infrastructure with little effort to create a constantly evolving network.

7 Conclusion

Our main conclusion is that an attack infrastructure can remediate most weaknesses of a cyber weapon, as it supports the deployment of new versions of a weapon. An analysis of attack tactics, techniques, and procedures by criminal gangs and malware developers confirms the importance of this infrastructure. A prerequisite for creating such an infrastructure is the ability to persist stealthily within adversary networks. This is confirmed by the wide adoption of living-off-the-land techniques in intrusions because subverting the behaviour of tools already installed on the target system, rather than downloading new ones, minimises the likelihood of detection. A second conclusion is the importance of dismantling the attack infrastructures of potential adversaries as a preliminary step to defend against cyber weapons.

References

1. Axelrod, R., Iliev, R.: Timing of cyber conflict. Proceedings of the National Academy of Sciences **111**(4), 1298–1303 (2014). https://doi.org/10.1073/pnas.1322638111
2. Boschetti, N., Gordon, N, G., Falco, G.: Space cybersecurity lessons learned from the viasat cyberattack. In: ASCEND 2022. p. 4380 (2022)
3. Boyarchuk, O., Mariani, S., Ortolani, S., Vigna, G.: Keeping up with the emotets: Tracking a multi-infrastructure botnet. Digital Threats: Research and Practice **4**(3), 41 (2023)
4. Caltagirone, S., Pendergast, A., Betz, C.: The diamond model of intrusion analysis. Tech. rep., Threat Connect (2013)
5. Cimpanu, C.: Risky bulletin: Six-years-old backdoor comes to life to hijack magento stores. https://risky.biz/risky-bulletin-six-years-old-backdoor-comes-to-life-to-hijack-magento-stores/ (2025)
6. Ebad, A., Darem, A., Abawajy, J, H.: Measuring software obfuscation quality–a systematic literature review. IEEE Access **9**, 99024–99038 (2021)
7. Farwell, J.P., Rohozinski, R.: Stuxnet and the future of cyber war. Survival **53**(1), 23–40 (2011)
8. Ghanbari, H., Koskinen, K., Wei, Y.: From solarwinds to kaseya: The rise of supply chain attacks in a digital world. Journal of Information Technology Teaching Cases (2024)
9. Google Threat Intelligence Group: Cybercrime: A multifaceted national security threat. https://cloud.google.com/blog/topics/threat-intelligence/cybercrime-multifaceted-national-security-threat (2025)
10. Healey, J.: Cyber operations and maschmeyer's "subversion trilemma". Lawfare, https://www.lawfaremedia.org/article/cyber-operations-and-maschmeyers-subversion-trilemma (2022)
11. Herr, T.: Prep: A framework for malware & cyber weapons. Journal of Information Warfare **13**(1), 87–106 (2014)
12. Huskaj, H, G., Iftimie, Ion, A., Wilson, R, L.: Designing attack infrastructure for offensive cyberspace operations. In: European Conference on Information Warfare and Security, ECCWS (2020)
13. Insikt Group: Threat actors leverage internet services to enhance data theft and weaken security defenses. Tech. rep. (2023)
14. Insikt Group: 2024 malicious infrastructure report. Tech. rep. (2025)
15. Kareem, K., et al.: Understanding the defence of operational technology (ot) systems: A comparison of lockheed martin's cyber kill chain, mitre att&ck framework, and diamond model. In: The International Conference on Computing, Communication, Cybersecurity & AI. Springer Nature Switzerland (2024)
16. Khattak, S., et al.: A taxonomy of botnet behavior, detection, and defense. IEEE communications surveys & tutorials **16**(2), 898–924 (2013)
17. Libicki, M.C.: Cyberspace is not a warfighting domain. A Journal of Law and Policy for the Information Society **8**(2), 326 (2012)
18. Lindsay, J.R.: Stuxnet and the limits of cyber warfare. Security Studies **22**(3), 365–404 (2013). https://doi.org/10.1080/09636412.2013.816122
19. Mandiant: The gru's destructive playbook. Tech. rep. (2023)
20. Mandiant: Ioc extinction? china-nexus cyber espionage actors use orb networks to raise cost on defenders. Tech. rep. (May 2024)

21. Maschmeyer, L.: The subversive trilemma: Why cyber operations fall short of expectations. International Security **46**(2), 51–90 (2021). https://doi.org/10.1162/isec_a_00418
22. Maschmeyer, L.: Subverting skynet: The strategic promise of lethal autonomous weapons and the perils of exploitation. In: 14th International Conference on Cyber Conflict: Keep Moving! (CyCon). pp. 155–171. Tallinn, Estonia (2022). https://doi.org/10.23919/CyCon55549.2022.9811008
23. Maschmeyer, L.: Subversion: From Covert Operations to Cyber Conflict. Oxford University Press (2024)
24. Maschmeyer, L., Cavelty, M.D.: Goodbye cyberwar: Ukraine as reality check. Tech. Rep. 3, CSS Policy Perspectives (2022)
25. McEvoy, T.R., Wolthusen, S.D.: Paying the rent: A formal methods riposte to "living off the land" attacks. In: Oliva, G., Panzieri, S., Hämmerli, B., Pascucci, F., Faramondi, L. (eds.) Critical Information Infrastructures Security. Lecture Notes in Computer Science, vol. 15549. Springer (2024). https://doi.org/10.1007/978-3-031-84260-3_6
26. Milenkoski, A., Minier, J., Vögele, J, F., Smeets, M., Grossman, T.: Ransomware's new masters: How states are hijacking cybercrime. Tech. rep., Pharos Report No. 3, Virtual Routes (2025)
27. MITRE: Emotet. https://attack.mitre.org/software/S0367/
28. Nguyen D, D, A., Alain, P., Autrel, F., Bouabdallah, A., J, F., G, D.: How fast does malware leveraging eternalblue propagate? the case of wannacry and notpetya. In: IEEE 10th International Conference on Network Softwarization (NetSoft). pp. 399–404. Saint Louis, MO, USA (2024)
29. Rai, R.: Behavioural threat detection: Detecting living of land techniques. http://essay.utwente.nl/83610/ (2020)
30. Reinhold, T., Reuter, C.: Toward a cyber weapons assessment model—assessment of the technical features of malicious software. IEEE Transactions on Technology and Society **3**(3), 226–239 (2022)
31. Rid, T.: Cyber war will not take place. Journal of Strategic Studies **35**(1), 5–32 (2011). https://doi.org/10.1080/01402390.2011.608939
32. Rodriguez, M., et al.: A framework for evaluating emerging cyberattack capabilities of ai. arXiv preprint arXiv:2503.11917 (2025)
33. Slovik, J.: Spyware stealer locker wiper. lockergaga revisited. Tech. rep., Dragos Inc. (2020)
34. Smeets, M.: A matter of time: On the transitory nature of cyberweapons. Journal of Strategic Studies **41**(1–2), 6–32 (2017). https://doi.org/10.1080/01402390.2017.1288107
35. Talib, M, A., Nasir, Q., Nassif, A, B., Mokhamed, T., et al.: Apt beaconing detection: A systematic review. Computers & Security **122**, 102875 (2022)
36. Vatsyayan, V., et al.: A detailed investigation of popular attacks on cyber physical systems. In: Cyber Security Applications for Industry 4.0. Chapman and Hall/CRC (2022)
37. VMware Threat Analysis Unit: Emotet exposed: A look inside the cybercriminal supply chain. Tech. rep. (2022)
38. Wilson, C.: Cyber weapons: 4 defining characteristics. GCN.com, https://www.route-fifty.com/cybersecurity/2015/06/cyber-weapons-4-defining-characteristics/287193/ (2015)