Options for Persistence of Cyberweapons

Carissa G. Hall

and Neil C. Rowe (presenter)

U.S. Naval Postgraduate School

Monterey, California, USA

ncrowe@nps.edu

Motivation

- A cyberweapon is weaponized software code that usually exploits flaws in software.
- Cyberweapons are only effective while the flaw still exists.
- Because of this, there is often only a short window of time when a particular cyberweapon can be used.
- We discuss how an attacker can increase the length of the window in several ways.
- Persistence is important to understand for both offense and defense.

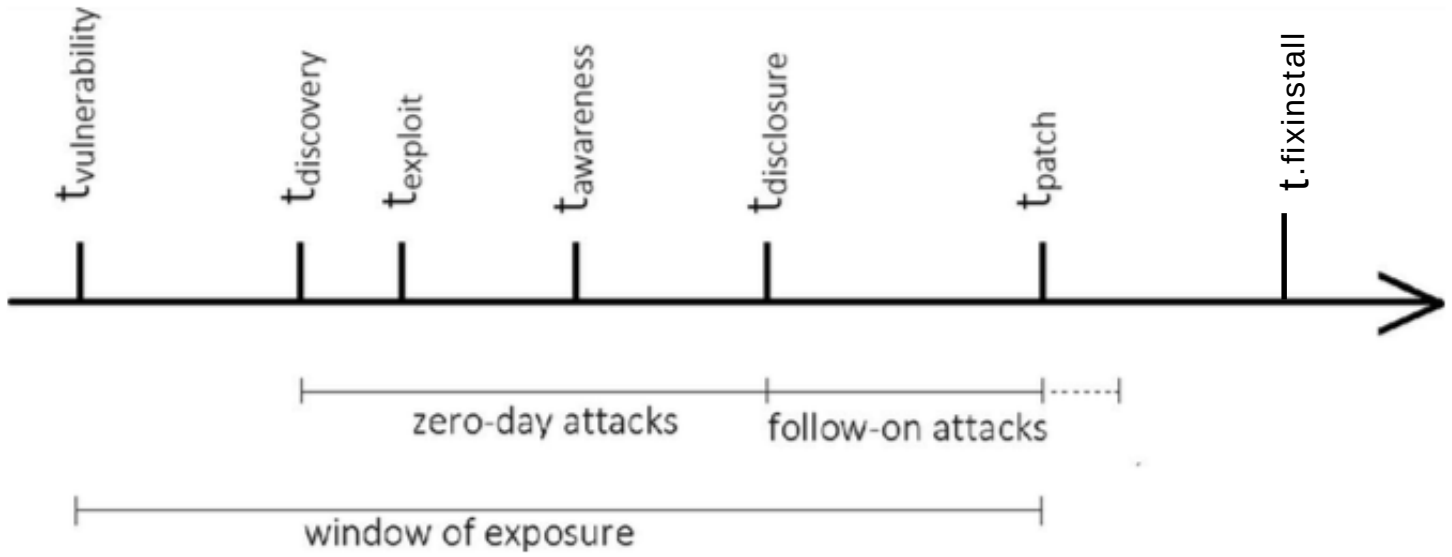
Cyberweapons don't last forever

- Since digital systems are designed to be reliable, it is hard to build a cyberweapon.
- The best hope is to find an unknown flaw in software, a "zero-day" vulnerability.
- The flaw may get fixed unexpectedly, rendering the cyberweapon useless.
- A successful attack leaves plenty of evidence about what happened, and its vulnerability can be patched within a week.
- If we want to stockpile a cyberweapon, or later reuse it, we need to find ways to extend its lifespan.

Some models of cyberweapons

- (Herr, 2014): A propagation method, an exploit, and a payload.
- (Arbaugh et al, 2000): Vulnerability, birth of exploit, discovery of exploit, disclosure of vulnerability, correction of vulnerability, publicity of exploit and vulnerability, scripting of countermeasures, and death of vulnerability.
- (Wilson et al, 2016): Parties as independent agents, small teams, large teams, and government organization. They can be offense or defense, and they can be for good or evil.
- (Shazad et al, 2012): Actions on discovery of a vulnerability:
  - Keep it secret and use it to attack.
  - Keep it secret and sell it.
  - Report it only to the software vendor.
  - Report it to the public directly.

The vulnerability time cycle



T-vulnerability: When the vulnerability manifested in the code
T-discovery: When someone discovered the vulnerability
T-exploit: When someone created an exploit for the vulnerability
T-awareness: When a code originator discovered something wrong regarding the vulnerability (their discovery)
T-disclosure: When vulnerability was publicly announced
T-patch: When a fix for the vulnerability is released
T-fixinstall: When a fix is installed on most computer systems

Estimating times in the vulnerability life cycle

- Time between vulnerability birth and discovery can vary greatly depending on the complexity of the vulnerability.
- Time between discovery and exploit depends on what the discover does. If they publicly disclose it, exploit creation and vendor awareness occur quickly; if they keep it secret, exploit creation occurs quickly, and vendor must wait for awareness or disclosure.
- Time between exploit and awareness depends how stealthy the exploit is. A stealthy exploit can be used for a while.
- Time between awareness and disclosure depends on the responsibility of the code originator. Commercial companies tend to announce quickly, with some exceptions.
- Time between disclosure and patch is usually a few days.
- Time between patch and installation is usually a week.

Persistence options for the cyberattacker

- Shortening time between vulnerability birth and discovery provides no direct benefit for the attacker.
- Shortening time between discovery and exploit provides a little benefit for the attacker, but not much.
- Lengthening time between exploit and awareness is very desirable for attackers. It can be done by making effects subtle, but most useful attacks intend unsubtle effects.
- Lengthening time between awareness and disclosure is possible by attacking code from an irresponsible originator or one with limited resources.
- Lengthening time between disclosure and patch is also due the code originator and hard for an attacker to control.
- Lengthening time between patch and installation can be increased by the attacker attacking something hard to update.

More about vulnerability and exploit awareness

- Odd behavior observed by an anomaly-based intrusion-detection system or network analysis can lead to malware discovery.
- Even not knowing the vulnerability, signatures can be created for odd behavior. Signatures allow for quick checking for it.
- Many organizations and companies are developing signatures all the time from observing network traffic. The more an exploit appears on the Internet, the more likely the originator will become aware of it.
- If an obvious attack occurs, signatures will be created within days and disseminated throughout the world.
- Signature creation can be modeled as a Poisson process.

Contributory negligence in cybersecurity

- A key problem in cybersecurity is failure to patch systems with known vulnerabilities and known fixes.
- This can be because:
  - Vendor negligence in supplying a patch, getting it to customers, or making it easy to install.
    - Sometimes Microsoft has been slow to provide a patch because they think it will break something else, like CVE-2017-0199 which Microsoft patched 9 months after discovery.
  - User negligence. Sometimes users do not set up for automatic updates or do not understand its importance.
  - Active interference with patch by the attacker: Hard to do.
  - Social engineering: People still fall victim to phishing and similar scams.

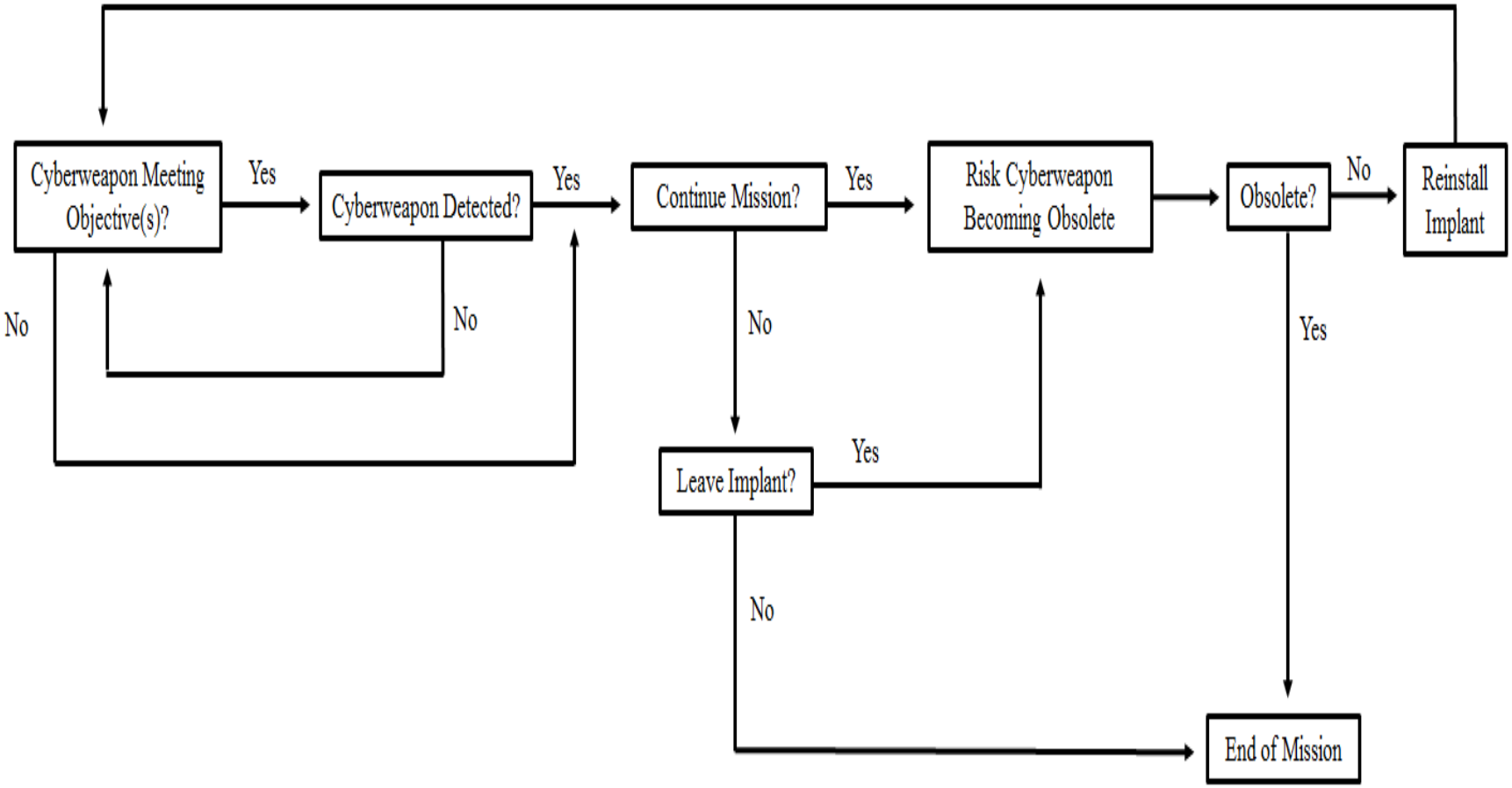
Hiding and camouflage of malware

- The time between deployment of an exploit and awareness can be increased by hiding it.
- An exploit can be hidden in someplace not often examined:
  - The operating system
  - Main memory after decryption from secondary storage
  - Hardware
- An exploit can also be hidden by camouflage:
  - Varying the details with polymorphic malware
  - Encrypting the malware

Survivability of cyberweapons

- It is useful to make an analogy to military aircraft survivability.
- A military aircraft needs to be able to handle multiple threats to its mission.
- Exploits similarly need to thwart various threats to their discovery and disabling.
- They can be more survivable if they have multiple techniques to accomplish their objectives, so that if one fails, they can try another.
- Stuxnet used multiple techniques to ensure it propagated well.

Visualization of cyberweapon survivability



Multipurpose persistent weapons

- The same persistent weapon may need to be used differently against different targets.
- The same persistent weapon may need to be used differently at different times. For instance:
  - Suppose a crisis escalates and we have an implant on an adversary's system.
  - The weapon may change its mission from exfiltration of intelligence to cyber-sabotage.
- Policy should be prepared for such changes.
- Currently U.S. law distinguishes Title 10 operations (military ones) from Title 50 (intelligence ones). Title 10 operations must be disclosed. Changing functions could change the legal basis.

Policy recommendations

Designers of cyberweapons have disjoint choices:

- Single-use cyberweapon:
  - Does not need to be stealthy.
  - Appropriate when it needs to make a point.
  - Appropriate to prevent a rare event.
- Potentially reusable cyberweapon:
  - Needs to be stealthy.
  - Needs innovative or difficult-to-analyze methods.
  - Needs multiple methods.
  - Appropriate when continuing harassment is needed.

Persistence and deterrence

- Our recent work looks at cyber deterrence.
- Generally, cyber deterrence by showing capabilities doesn't work as well as deterrence with conventional weapons:
  - You can't usually demonstrate a weapon without causing it to lose some or much of its effectiveness, since an observed attack can be analyzed to some extent.
  - Thus bluffing is easy with cyberweapons and announcing a capability doesn't have much deterrence.
  - Sanctions, criminal indictments, improving defenses, and mounting offensive cyber operations all have drawbacks as deterrents.
- But provably persistent weapons could be an effective deterrent, if they are hard to find and neutralize.