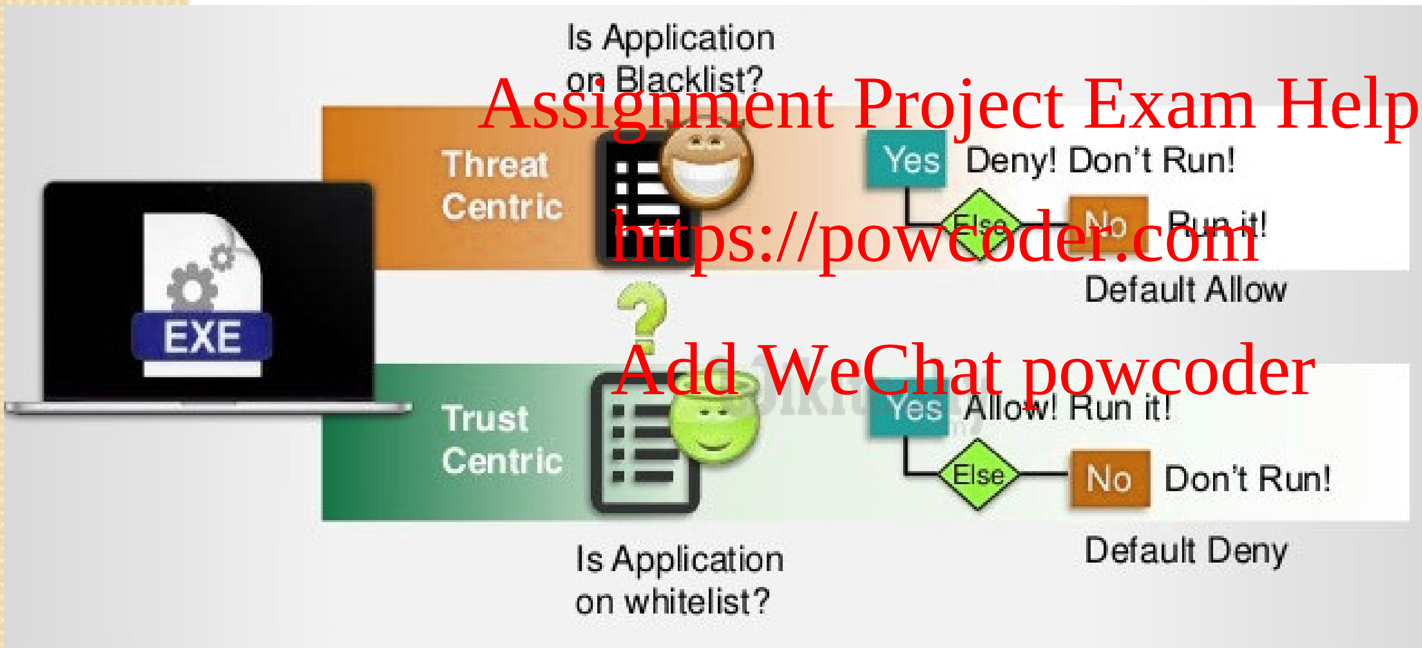
Threat Events: Software Attacks (cont.)

Blacklisting vs. Whitelisting



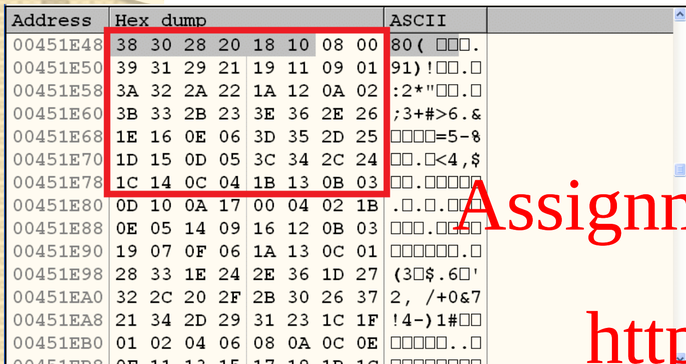
Blacklisting:
allow everything
block some
good for detecting
yesterday's threats

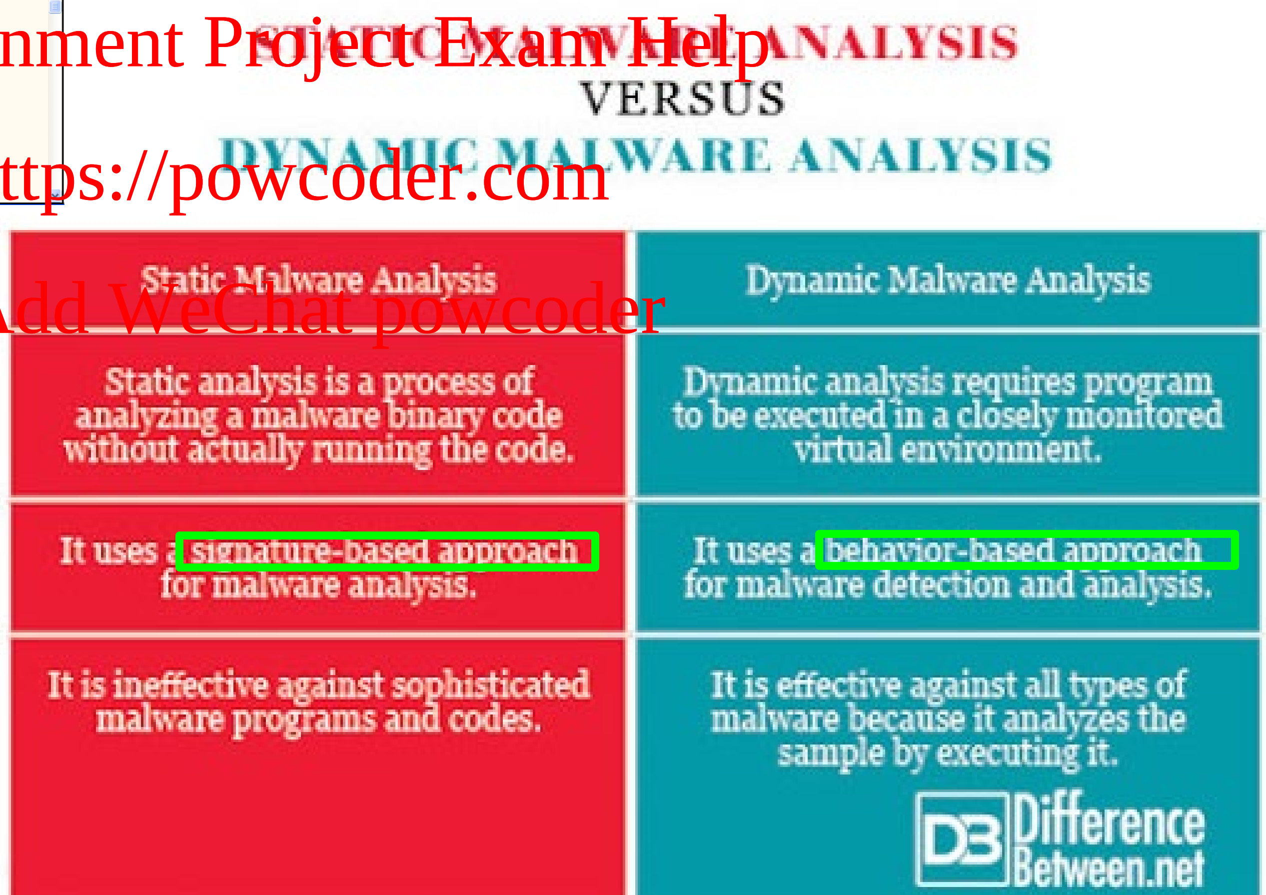
Whitelisting:
block everything
allow some
good for detecting
zero-day threats

https://www.wikitechy.com/interview-questions/networking/what-is-whitelist

Threat Events: Software Attacks (cont.)

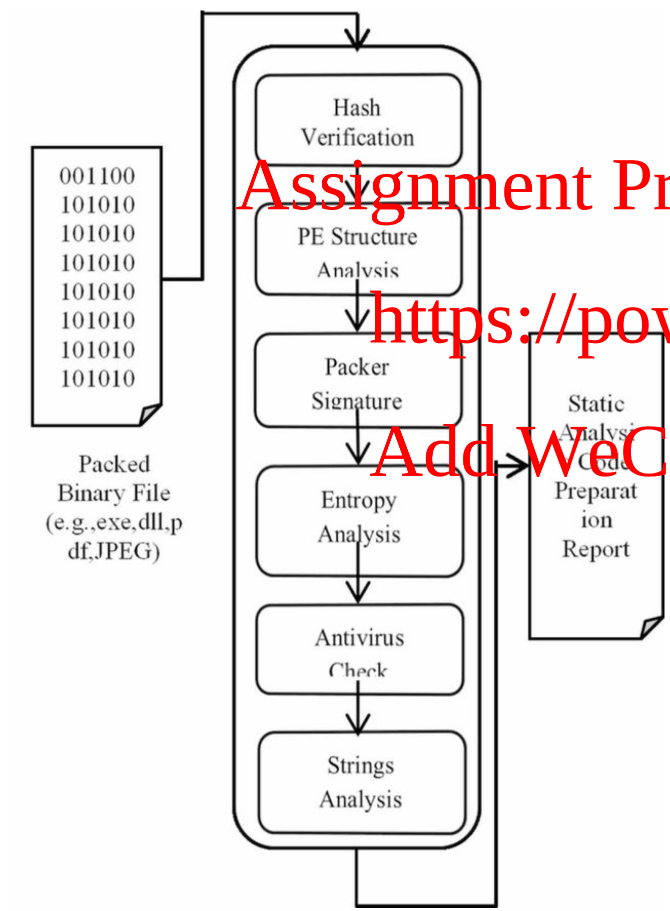
Types of Virus/Malware Analysis: Static vs. Dynamic



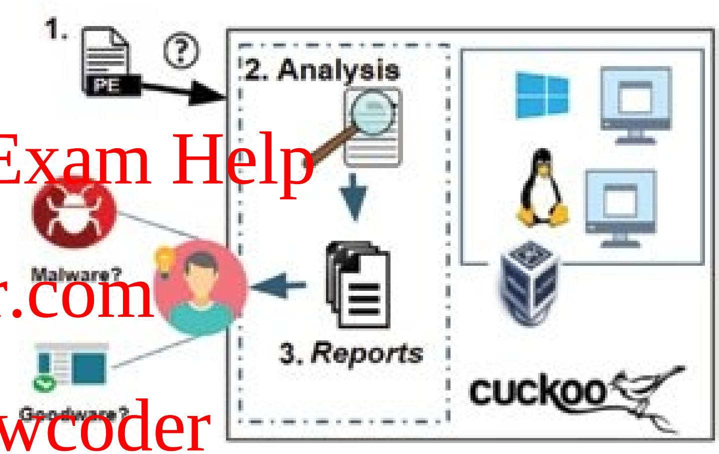


Threat Events: Software Attacks (cont.)

Static Malware Analysis



Dynamic Malware Analysis



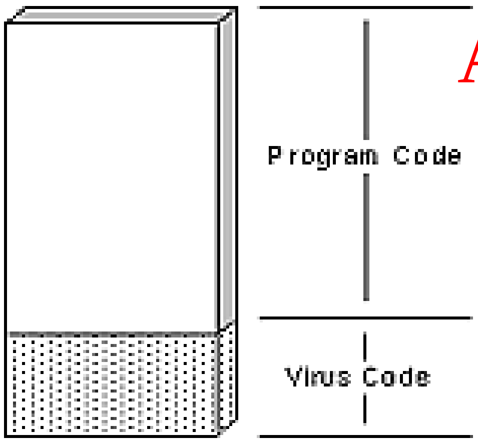
A sandbox typically provides a tightly controlled set of resources for guest programs to run in. **Network access, the ability to inspect the host system or read from input devices are usually <u>disallowed or heavily restricted</u>.**

https://www.researchgate.net/publication/332215777_A_Mathematical_Model_of_HMST_Model_on_Malware_Static_Analysis/figures?lo=1

https://www.researchgate.net/publication/329496012_Building_malware_classificators_usable_by_State_security_agencies/figures?lo=1

Threat Events: Software Attacks (cont.)

- **VIRUS**



- ✶ **classification of viruses by concealment strategy**

  i) polymorphic virus – completely mutates (changes its appearance) with every infection to avoid 'signature' (bit pattern) detection

  avoids static-analysis detection

  iv) metamorphic virus - mutates and changes its behavior with every infection

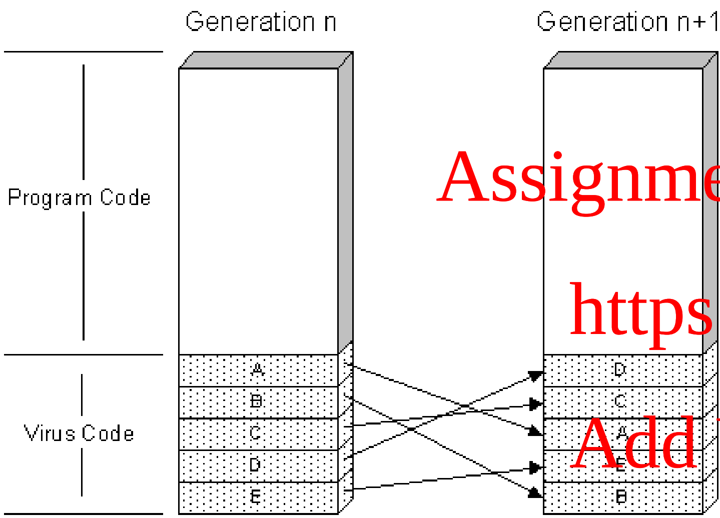
  avoids dynamic-analysis detection

  ii) encrypted virus - a portion of the virus creates a random key and encrypts the remainder - special case of polymorphic virus

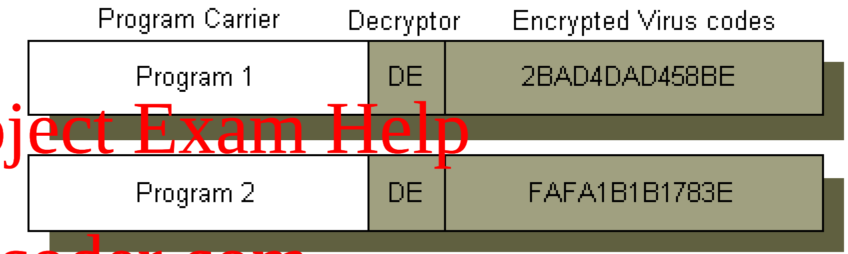
  iii) stealth virus - uses special techniques to conceal its presence on the OS
  - makes sure that 'last modified' date of host file remains unchanged
  - makes sure that the size of host file appears/ stays the same - aka cavity viruses

Threat Events: Software Attacks (cont.)

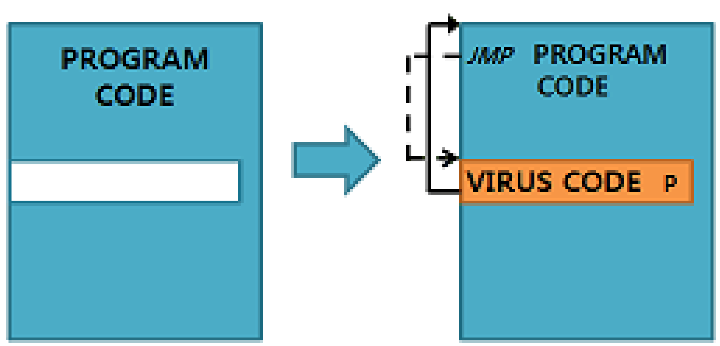
Polymorphic Virus



Encrypted Virus



Stealth (Cavity) Virus



Threat Events: Software Attacks (cont.)

- **WORM** – malware actively seeks out more machines to infect and then each infected machine serves as an automated launching pad for attacks on other machines

  - worms exploit software vulnerabilities in client or server programs to gain access to a new system

    (worm = power of virus + convenience of Internet)

  - IMPORTANT: viruses vs. worms
    - viruses need a carrier medium (document or program to 'attach' itself to) and then require user action to propagate
    - worms do not always need a carrier (can some-times 'move' on their own), are typically spread through the Internet, and NEVER rely on user action TO REPLICATE

Threat Events: Software Attacks (cont.)

- **WORM**

  - classification of worms by replication strategy

Propagates on its own, but requires user action to activate.

1) electronic mail or instant messaging - worm emails a copy of itself to other systems, or sends itself as an attachment via an instant message service

Propagates and activates on its own.

2) file sharing - worm copies itself on removable media such as USB drives; it, then, executes when the drive is connected to another system

3) remote login capability - worm logs onto a remote system as a user and then uses commands to copy itself from one system to another

4) remote file access or transfer capability - worm uses a remote file access or transfer service to another system to copy itself

etc. ....

Threat Events: Software Attacks (cont.)

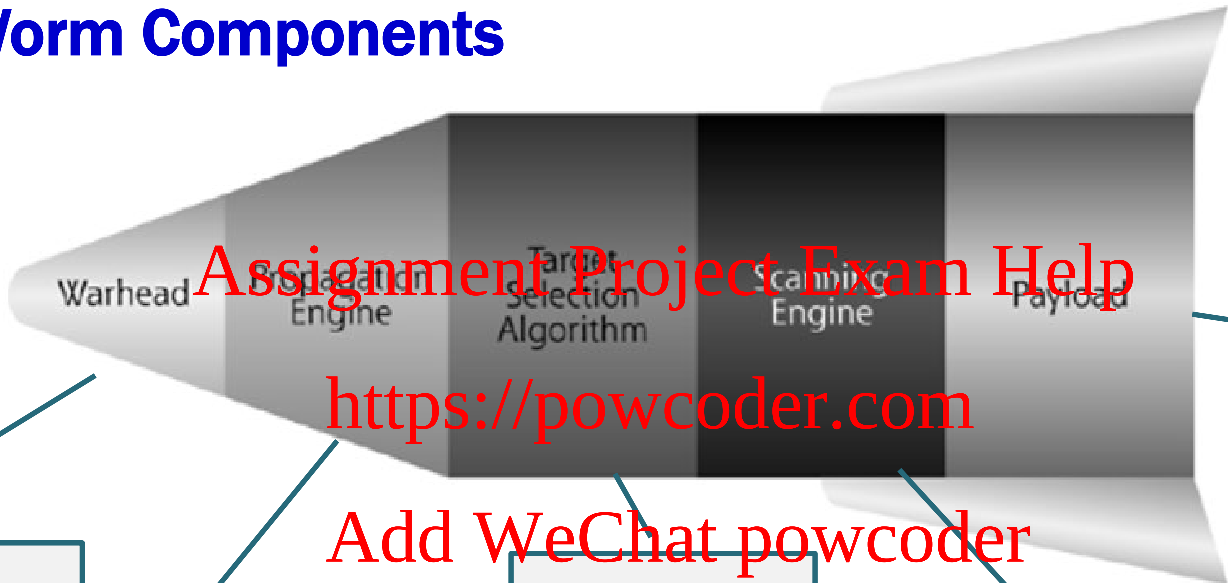
<u>Example</u>: USB Virus vs. USB Worm

VIRUS: Malware 'sits' inside a 'carrier' (program/document) and requires one user to manually move the carrier 'onto' and another user to move it 'from' a USB and click on it.

Worm: Malware on its own infects the USB, and (when plugged into a new machine) on its own moves from the USB onto the new victim machine.

Threat Events: Software Attacks (cont.)

Worm Components



Methods worms use to first gain access to the victim machine:
- drive-by-download
- email
- file sharing
etc.

Methods worms use to transfer the rest of its body to the target:
- file transfer
- HTTP
etc.

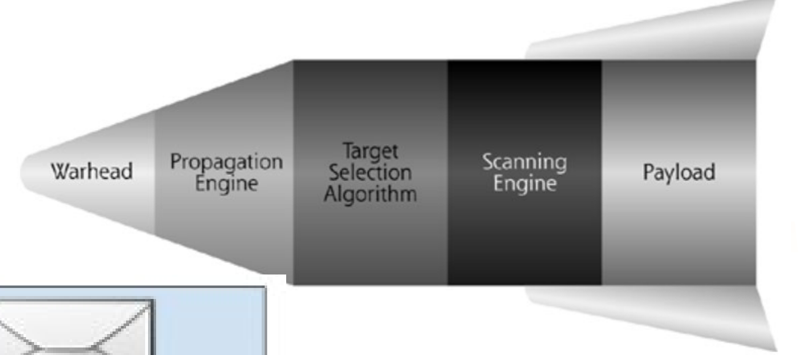
Once the worm is running on the victim machine it starts looking for new victims to attack
- email address
- host lists
- different IPs targets
etc.

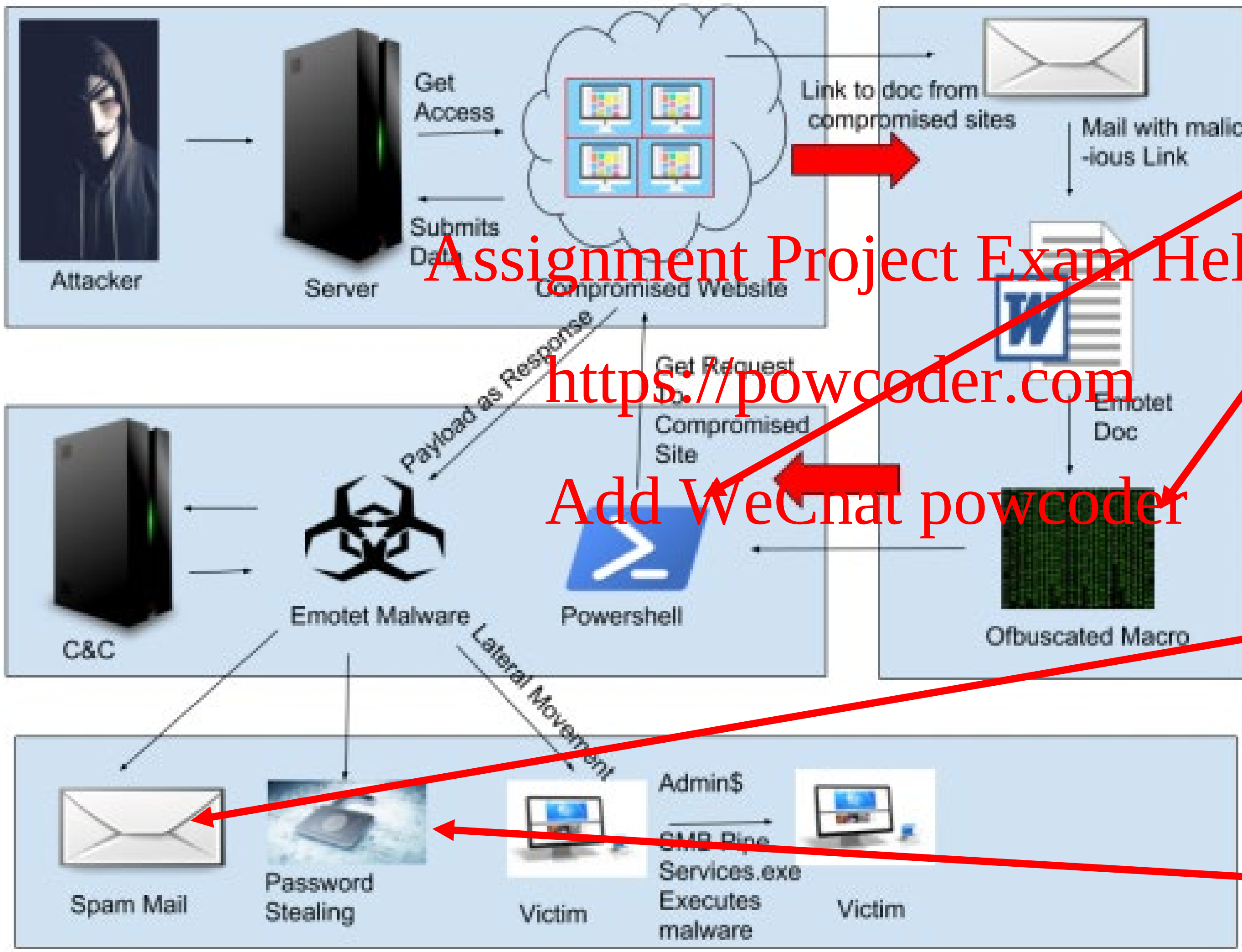
Using addresses generated by the target engine, the worm actively scans across the network to determine suitable victims

Chunk of code designed to implement some specific action on behalf of the attacker on a target system. It is what the worm does when it gets to a target …
- opening a backdoor
- planting a DDoS bot
- performing a complex math operation (e.g., cryptominer)

https://userpages.umbc.edu/~dgorin1/432/worms.htm

Emotet





Propagation Engine

Warhead - small hard-to-detect piece of code

Target Selection Algorithm + Scanning Engine

Payload

https://blogs.quickheal.com/beware-your-website-might-be-delivering-emotet-malware/

Threat Events: Software Attacks (cont.)

- WORM
  - classification of worms by target discovery
    - a) random - each compromised host probes random addresses in IP address space - fast process, but 1) unknown results (many machines may not be vulnerable), 2) some machine may already infected
    - b) hit list - the attacker compiles a long list of potentially vulnerable machines, each infected machine uses a part of this list - time consuming
    - c) topological - worm uses information contained on the infected machine to find more hosts to scan - e.g., worms infecting/exploiting P2P applications
    - d) local subnet - worm uses the subnet address to find other vulnerable machine on the same network (works well against firewall-protection)

Threat Events: Software Attacks (cont.)

Ideally, we would want to 'catch' a worm while in Slow Start phase ...

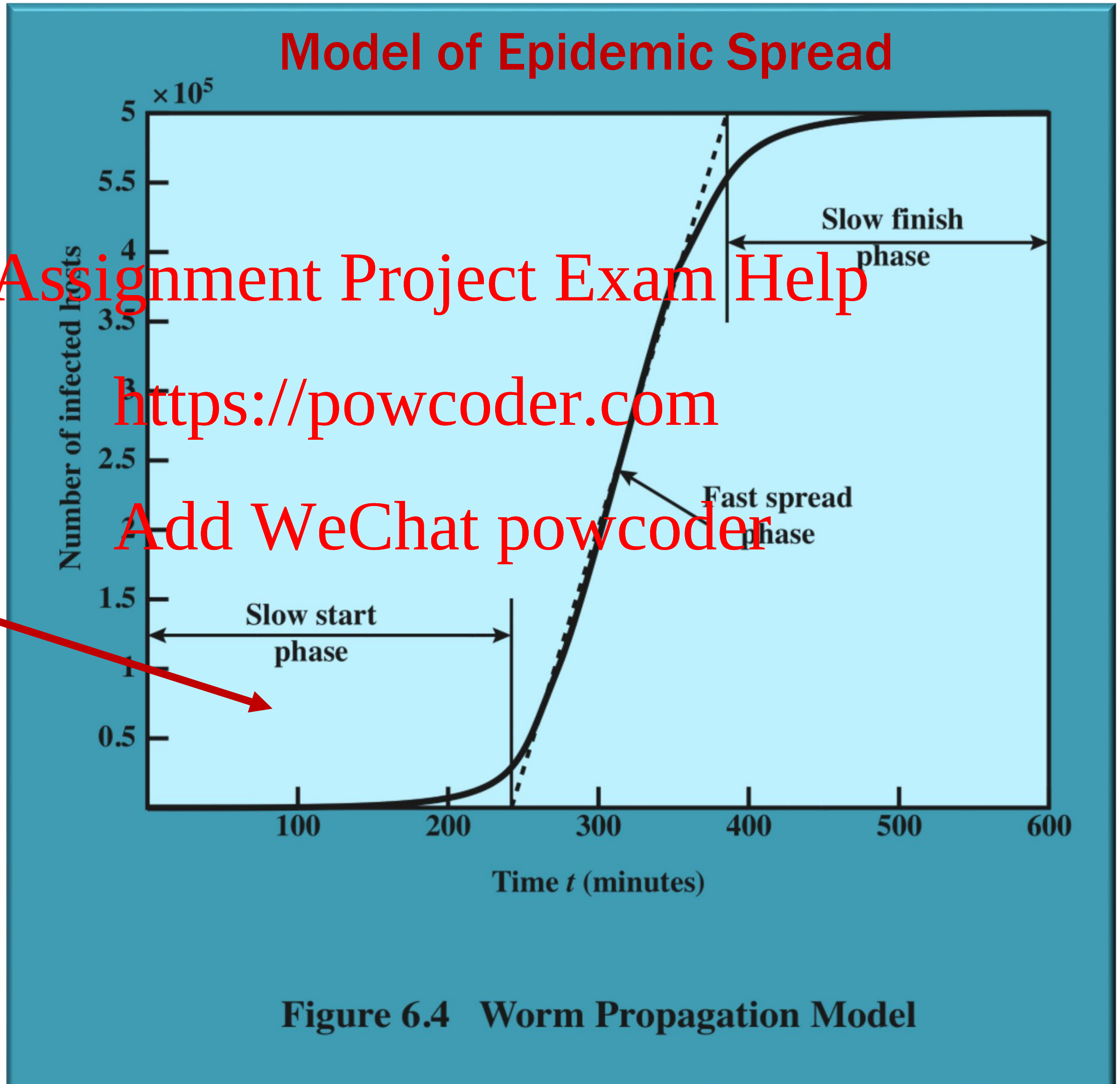


Figure 6.4 Worm Propagation Model

SLOW FINISH: most vulnerable machines have been infected

Example: Worm propagation …

Consider a network consisting of N machines and a worm that uses 'local network' propagation model. In particular, at time t=0, the worm has infected only 1 machine. In each subsequent minute, every infected machine contacts and successfully infects k=2 other machines on the same network. (You can also ssume:

1) All the machines in this network are vulnerable to the given worm.
2) The worm is 'smart' so that an infected machine never tries to infect another infected machine.)

If N = 200, how many minutes does it take to infect all the machines in the system?

Solution

1st minute: 1 old + 2 new infected = 3 infected machines

2nd minute: 3 old + 3*2 new infected = 9 infected machines

3rd minute: 9 old + 9*2 new infected = 27 infected machines

4th minute: 27 old + 27*2 new infected = 81 infected machines

5th minute: 81 old + 81*2 new infected = 243 infected machines

Threat Events: Software Attacks (cont.)

- **WORM**
  - state of worm technology

    i) multiplatform - target a variety of platforms / OSs

    ii) multi-exploit - penetrate systems in a variety of ways (through email, browsers, file sharing, ...)

    iii) ultrafast spreading - use various techniques to to identify as many vulnerable machines in a short period of time

    iv) polymorphic

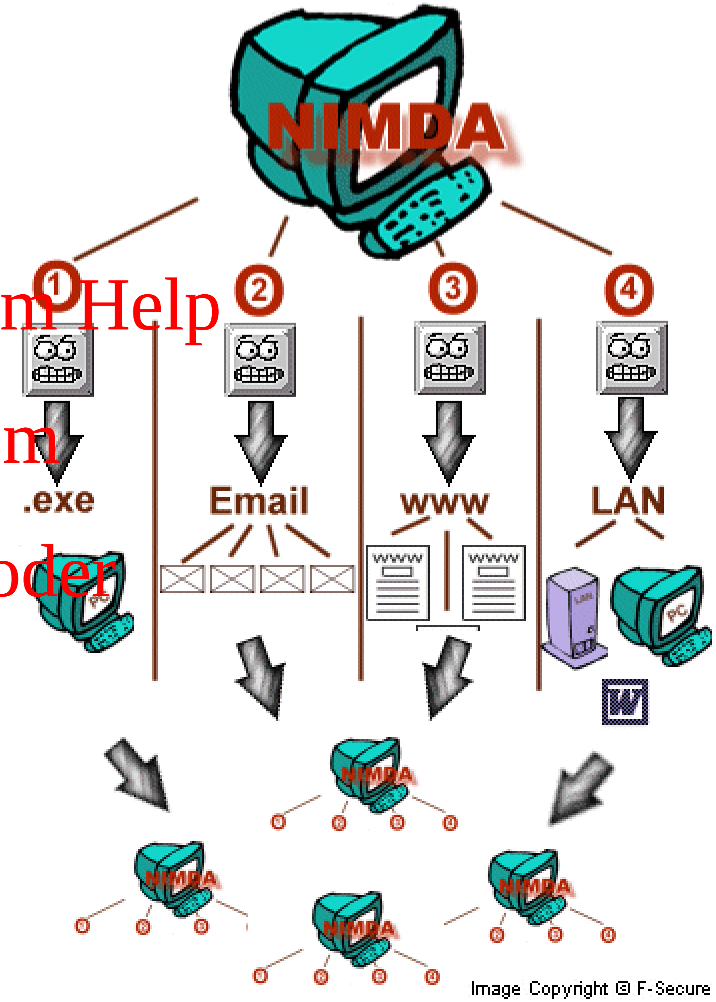
    v) metamorphic

    vi) multi 'transport vehicle' - can carry a variety of payloads (rootkits, spam generators, bots, etc.)

    vii) zero-day exploit - try to exploit new/unknown vulnerabilities

Threat Events: Software Attacks (cont.)

- **Nimda (2001)** – first **multi-exploit** worm – used 5 different infection paths:
  - * via email
  - * via browsing of compromised web sites – an injected java-script would allow the downloading of Nimda
  - * via open network shares on LANs
  - * via exploiting of vulnerabilities in Microsoft's IIS server
  - * via back doors left behind by the Code Red worms



Nimda cost an estimated $635 million in damages.

http://www.f-secure.com/v-descs/nimda.shtml

http://www.di.unisa.it/~ads/corso-security/www/CORSO-0102/NIMDA/link_locali/nimda-update-sept27.pdf

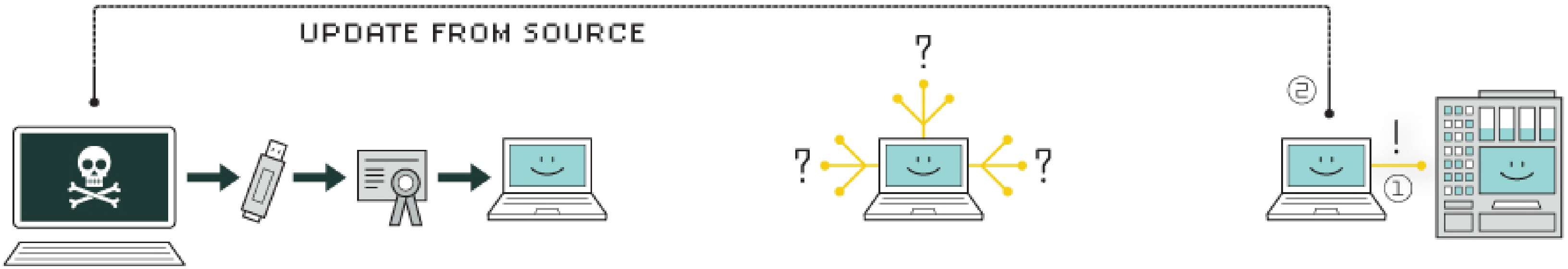
http://www.itsecurity.com/features/10-worst-virus-attacks-111207/

Threat Events: Software Attacks (cont.)

- **Stuxnet (2010)** – a highly sophisticated worm that used a variety of advanced techniques to spread, including:
  - by the use of shared infected USB drives (spreads even between computers that are not connected to the Internet);
  - by connecting to systems using a default database password;
  - by searching for unprotected administrative shares of systems on the LAN; …

  While it was programmed to spread from system to system, it was actually searching for a very specific type of system to execute – programmable logic controller (PLC) system made by Siemens and run on devices that control and monitor industrial processes. When it found such a system, it executed a series of actions designed to destroy centrifuges attached to the Siemens controller.

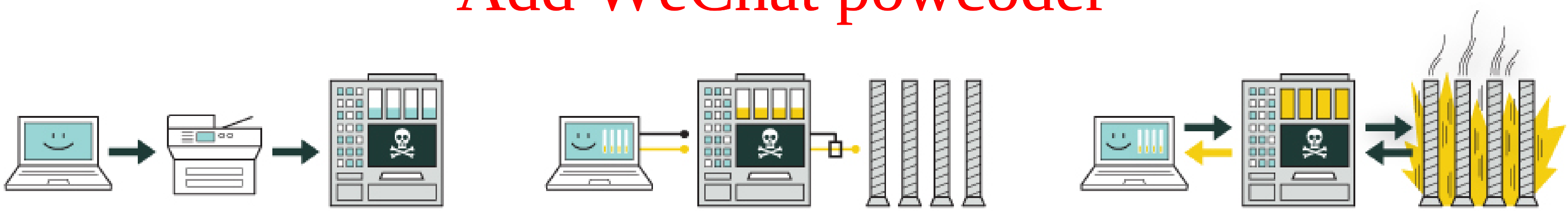
HOW STUXNET WORKED



1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn't a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.



4. compromise
The worm then compromises the target system's logic controllers, exploiting "zero day" vulnerabilities-software weaknesses that haven't been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won't know what's going wrong until it's too late to do anything about it.

http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

Stuxnet

https://www.youtube.com/watch?v=nEsNnwZpXrk

https://www.youtube.com/watch?v=LqDqD1tpl_E

OPTIONAL:

https://www.youtube.com/watch?v=oz585G-6NBA

https://www.youtube.com/watch?v=SAy46DhWW8Y