II- Malware

Objectives

At the end of this lesson, students will be able to:

- Understand and explain malware lifecycle
- Describe the types of malware
- Describe and illustrate the execution steps of viruses
- Classify viruses
- Explain social engineering techniques
- Describe DDOS attacks

Prerequisites

To be able to understand this lesson, students need notions on :

- Algorithms
- Assembler programming
- Network architecture

Keywords

MALWARE, DDOS, VIRUS, CREATION, GESTATION
REPRODUCTION, INFECTION, DISCOVERY,
ASSIMILATION, ELIMINATION VIRUS
WORM, TROJAN HORSE
MOBILE CODE, DOWNLOADERS
LOGIC BOMB, BACKDOOR (TRAPDOOR), EXPLOITS,
AUTO-ROOTER, KIT, SPAMMER PROGRAMS,
FLOODERS, ROOTKIT, ZOMBIE, BOT, SPYWARE,
ADWARE, KEYLOGGERS, TRIGGER,
PAYLOAD, PREMIUM RATE SMS
UPDATE ATTACK, DRIVE BY DOWNLOADS,
REPACKAGING, SOCIAL ENGINEERING,
DIRECT DDOS, REFLECTOR DDOS, DISTRIBUTED
SYN FLOOD ATTACK, DISTRIBUTED ICMP ATTACK

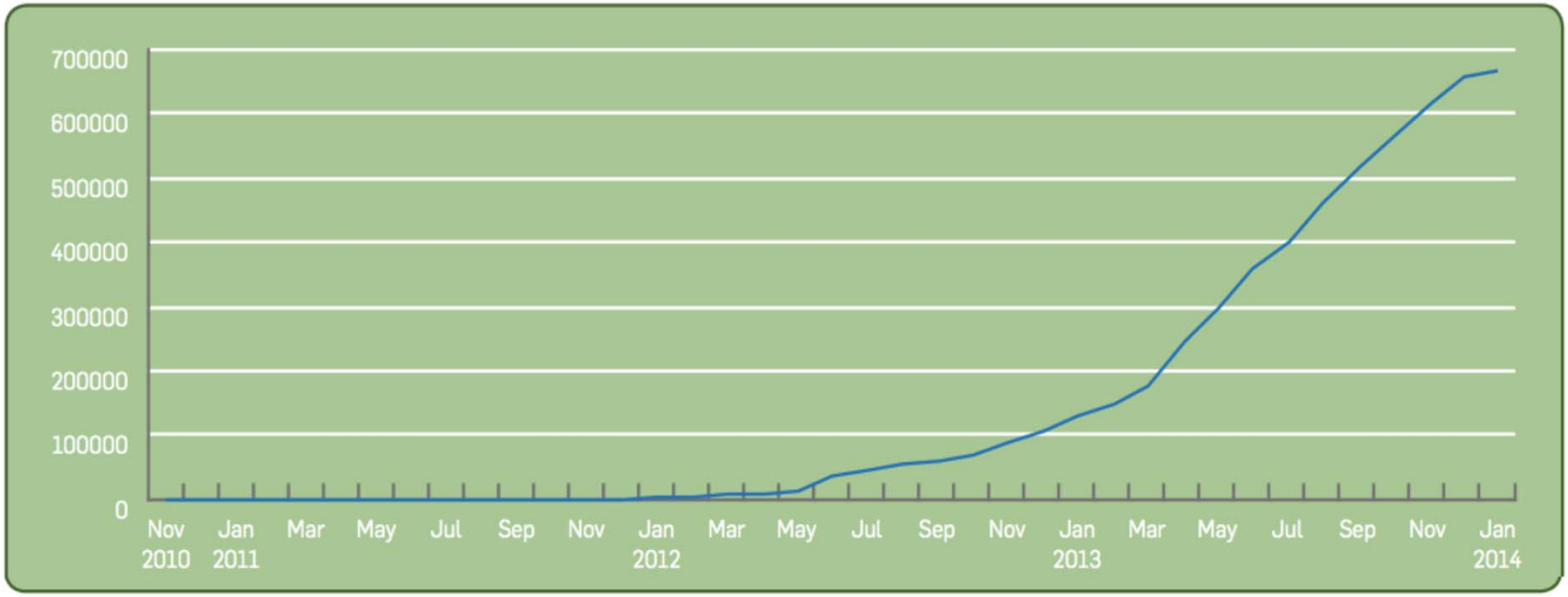
Literature

Kizza, J. M. : Guide to Computer Network Security, Third Edition. Computer Communications and Networks. Springer 2015
Stallings, W.: Cryptography and Network Security: Principles and Practice, Sixth Edition. Pearson, USA 2014

Further readings

1. Malicious Software

Malware = Malicious Software = An application that can be used to compromise an operation of a device, steal data, bypass access controls or otherwise cause damage on the host terminal.



2. Malware Lifecycle

- **Creation** : The programmer designs and implements all malicious code that will be included in the malware

- **Gestation**: Stage during which the malicious application infiltrates and settles in the system that it wants to infect. It remains inactive throughout this stage.

- **Reproduction or infection**: The malware reproduces a significant number of times before manifesting in this phase. The author of the malware seeks to remotely control devices and access private data.
- Activation: Some malware activates their destruction routine when certain conditions are satisfied (internal countdown reaches for example). The activation can also be done remotely. The purpose of this phase is to appropriate gradually all device resources.
- **Discovery**: The user notices strange behaviour and suspects the presence of a malicious application. This strange behaviour may include performance losses, current changes in the Web browser home page or the unavailability of certain system functions. Anti-viruses often assist the user in detecting malicious actions in sending alerts to the device owner. However, the furtive character of certain malware may extend, even complicate this phase.

- **Assimilation**: Antiviruses update their virus database after the discovery of new malware. If possible, a fix or antidote is also proposed to eliminate this threat.

- **Elimination**: the phase when the antivirus discovering the malware prompts the user to remove it. It marks the death of the malware

3. Terminology (1)

Virus: replicate itself into other executable code

Worm: run independently and can propagate a complete working version of itself onto other hosts on a network

Trojan Horse: A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program

Mobile code: Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.

Logic bomb: A program inserted into software by an intruder A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act

Backdoor (trapdoor): Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality.

Exploits: Code specific to a single vulnerability or set of vulnerabilities.

Downloaders: Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.

4. Terminology (2)

Auto-rooter: Malicious hacker tools used to break into new machines remotely

Kit (virus generator): Set of tools for generating new viruses automatically.

Spammer programs: Used to send large volumes of unwanted e-mail.

Flooders: Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.

Keyloggers: Captures keystrokes on a compromised system.

Rootkit: Set of hacker tools used after attacker has broken into a computer system and gained root-level access.

Zombie, bot: Program activated on an infected machine that is activated to launch attacks on other machines.

Spyware: Software that collects information from a computer and transmits it to another system.

Adware: Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

5. Comparison between categories of malware

Viruses, logic bombs and backdoors : those that need a host program referred to as parasitic, are essentially fragments of programs that cannot exist independently of Some actual application program, utility, or system program

Worms and bot programs: Independent malware is a self-contained program that can be scheduled and run by the operating system

Logic bombs, backdoors, and bot programs do not replicate. They programs or fragments of programs that are activated by a trigger.

Viruses and worms are program fragment s or independent programs that, when executed, may produce one or more copies of itself to be activated later on the same system or some other system.

6. Virus (1)

A **virus** can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs that is allowed by the privileges of the current user.

A computer virus has three parts

- **Infection mechanism:** a virus spreads, enabling it to replicate
- **Trigger:** The event or condition that determines when the payload is activated or delivered.
- **Payload:** What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity

A virus can be **prepended** or **postpended** to an **executable program**, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, **will first execute the virus code and then execute the original code of the program.**

7. Virus (2)

```
     program V :=

{goto main;
     1234567;

     subroutine infect-executable :=
          {loop:
          file := get-random-executable-file;
          if (first-line-of-file = 1234567)
             then goto loop
             else prepend V to file; }

     subroutine do-damage :=
          {whatever damage is to be done}

     subroutine trigger-pulled :=
          {return true if some condition holds}

main:    main-program :=
          {infect-executable;
          if trigger-pulled then do-damage;
          goto next;}
next:

}
```

Simple virus

```
     program CV :=

{goto main;
     01234567;

     subroutine infect-executable :=
               {loop:
                    file := get-random-executable-file;
               if (first-line-of-file = 01234567) then goto loop;
          (1)  compress file;
          (2)  prepend CV to file;
          }

main:    main-program :=
           {if ask-permission then infect-executable;
          (3)  uncompress rest-of-file;
          (4)  run uncompressed file;}
          }
```

Compressed virus

A way to thwart such a simple means of detecting a virus is to compress the executable file so that both the infected and uninfected versions are of identical length

Easily detected because an infected version of a program is longer than the corresponding uninfected one

8. Compressed Virus

We assume that program P1 is infected with the virus CV. When this program is invoked, control passes to its virus, which performs the following steps:

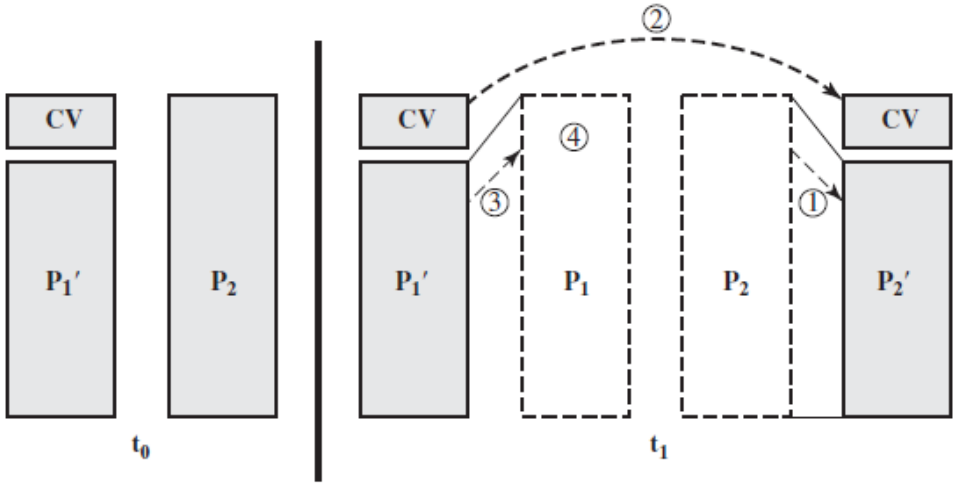
1. For each uninfected file P2 that is found, the virus first compresses that file to produce , which is shorter than the original program by the size of the virus.

2. A copy of the virus is prepended to the compressed program.

3. The compressed version of the original infected program, , is uncompressed.

4. The uncompressed original program is executed.

Propagation



Compressed virus

9. Virus Classification

By target

Boot sector infector
Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus

File infector
Infects files that the operating system or shell consider to be executable

Macro virus
Infects files with macro code that is interpreted by an application.

By concealment strategy

Encrypted virus
A portion of the virus creates a random encryption key and encrypts the remainder of the virus. The key is stored with the virus.

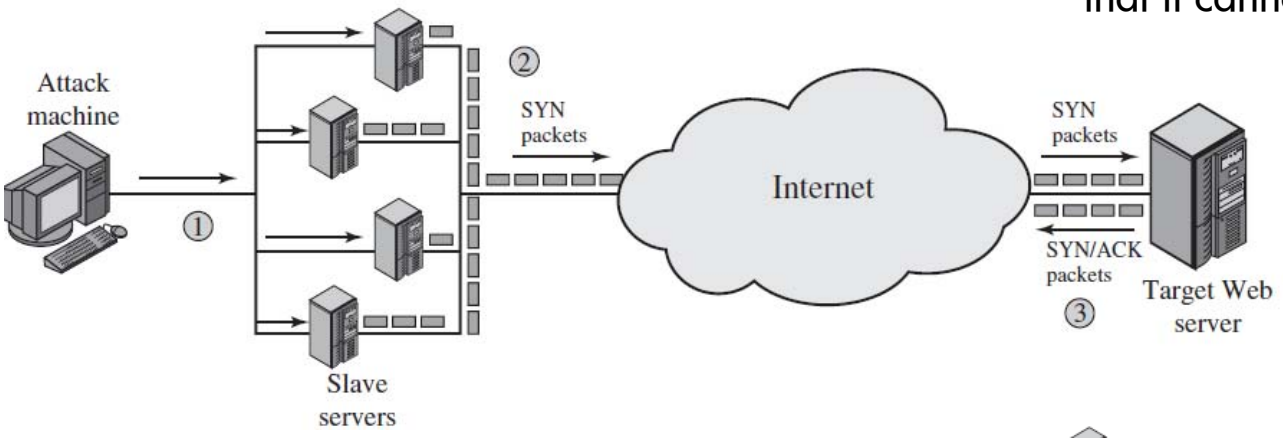
Polymorphic virus
A virus that mutates with every infection, making detection by the "signature" of the virus impossible

Stealth virus
A form of virus explicitly designed to hide itself from detection by antivirus software

Metamorphic virus
metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection

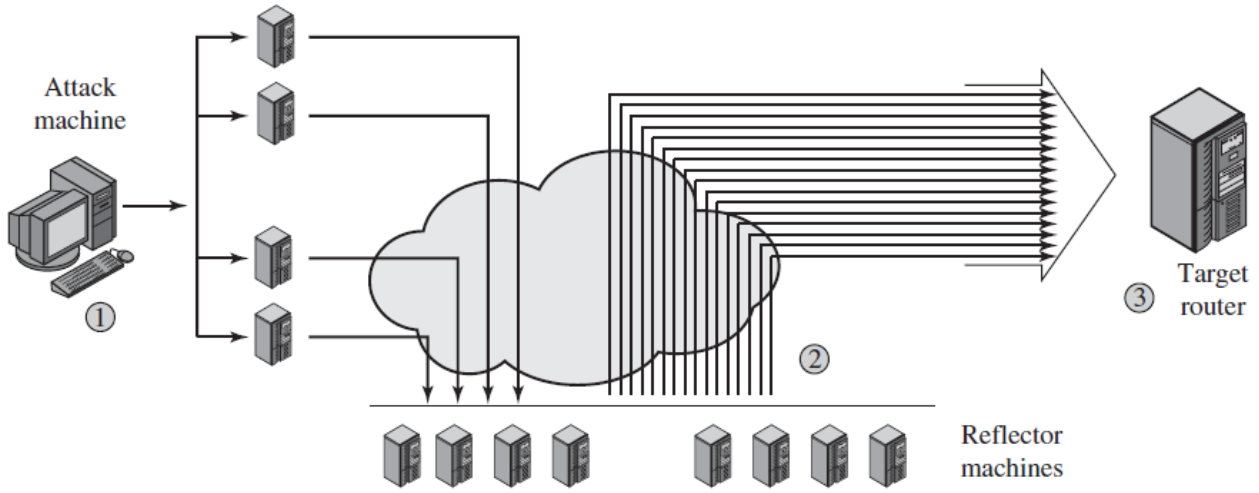
10. DDOS Attacks (1)

A DDoS attack attempts to consume the target's resources so that it cannot provide service.

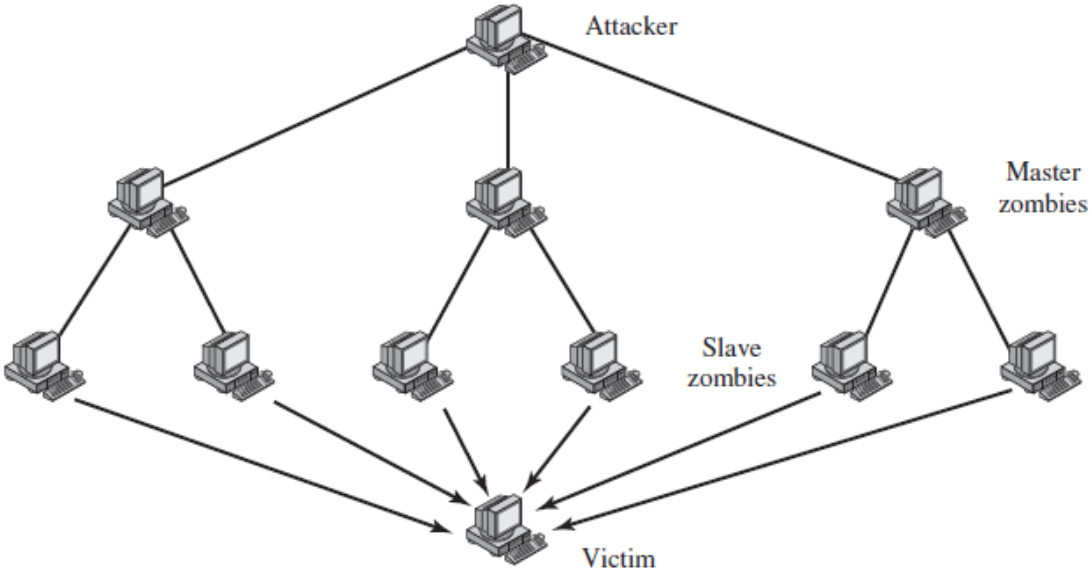


Distributed SYN flood attack

Distributed ICMP attack



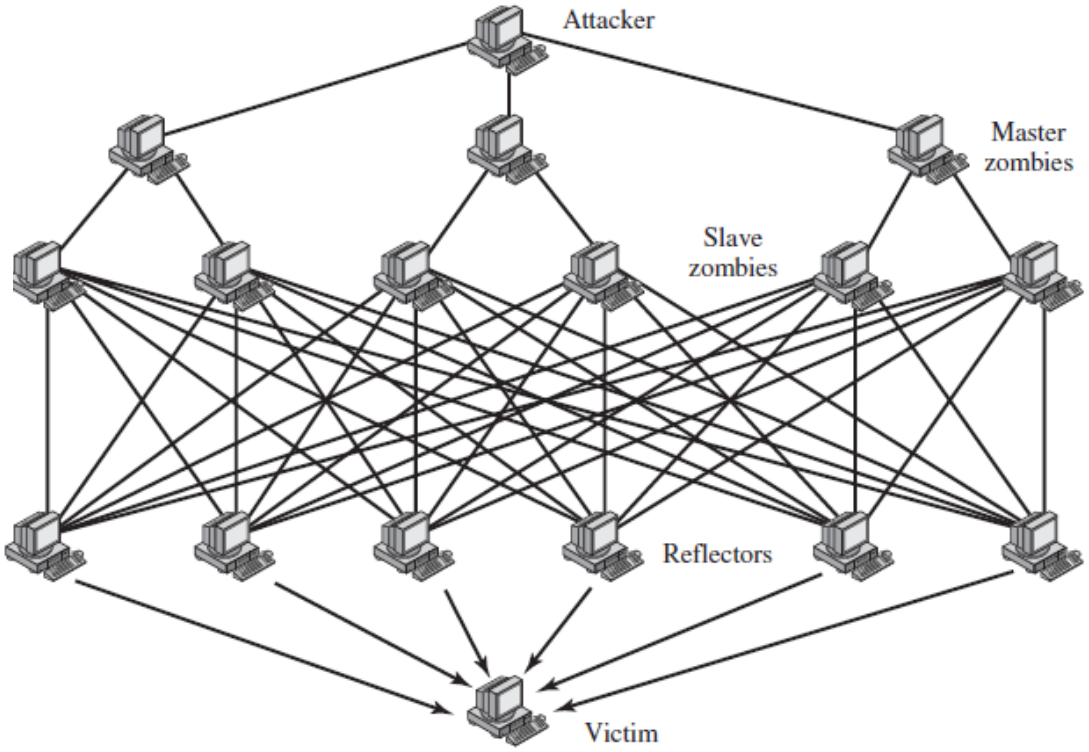
11. DDOS Attacks (2)



Direct DDoS

The attacker coordinates and triggers the master zombies, which in turn coordinate and trigger the slave zombies.

The slave zombies construct packets requiring a response that contains the target's IP address as the source IP address in the packet's IP header



Reflector DDoS

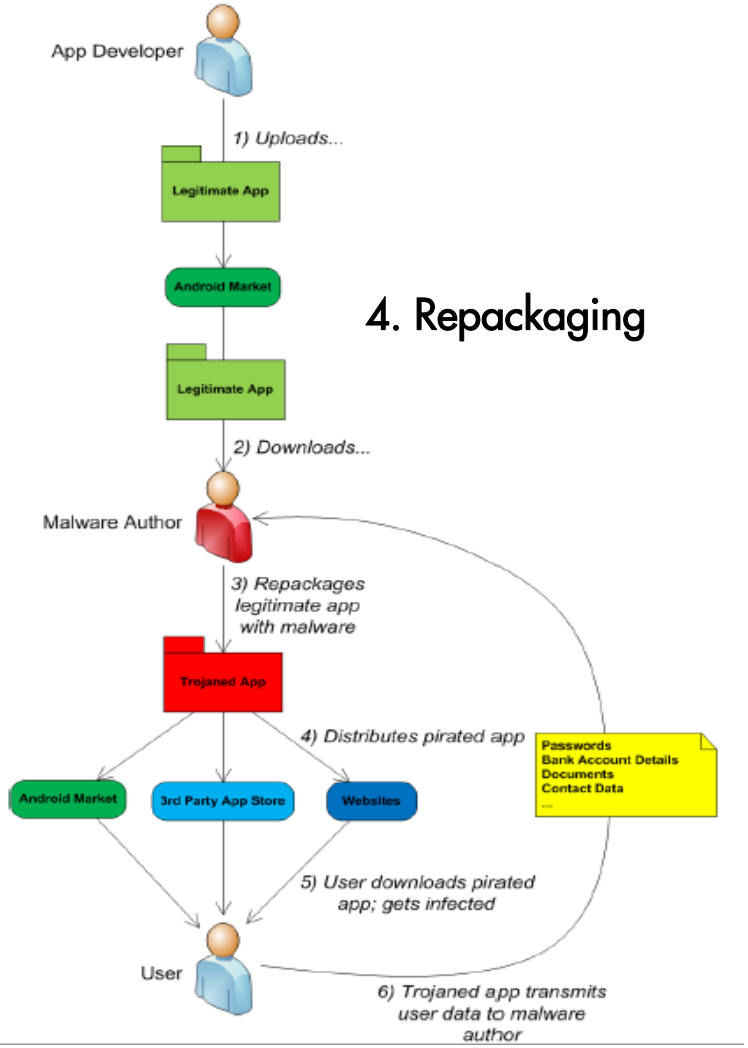
12. Social engineering

1. Premium Rate SMS

2. Update Attack

3. Drive by Downloads

4. Repackaging



13. Questions and Exercises (1)

- What is CVE? (Go to the website and find examples of vulnerabilities and exploits)
- What is CERT? What is the Cameroonian representattive? (Go to the website and find what is doing there)
- Give examples of malware
- Design an automata that involves different malware.
- Give some contermeasures for virus and worms
- Give some contermeasures for DDoS
- What is the role of compression in the operation of a virus?
- What is the role of encryption in the operation of a virus?
- What are typical phases of operation of a virus or worm?
- What is a digital immune system?
- How does behavior-blocking software work?
- In general terms, how does a worm propagate?
- Describe some worm countermeasures.
- What is a DDoS?

14. Questions and Exercises (2)

1.Consider the following fragment:

```
legitimate code
if data is Friday the 13th;
crash_computer();
legitimate code
```

What type of malicious software is this?

3. Consider the following fragment in an authentication program:

```
username = read_username();
password = read_password();
if username is "133t h4ck0r"
return ALLOW_LOGIN;
if username and password are valid
return ALLOW_LOGIN
else return DENY_LOGIN
```

What type of malicious software is this?

2. The following code fragments show a sequence of virus instructions and a metamorphic version of the virus. Describe the effect produced by the metamorphic code.

Original Code	Metamorphic Code

mov eax, 5
add eax, ebx
call [eax]
	mov eax, 5
	push ecx
	pop ecx
	add eax, ebx
	swap eax, ebx
	swap ebx, eax
	call [eax]
	nop

15. Questions and Exercises (3)

The point of this problem is to demonstrate the type of puzzles that must be solved in the design of malicious code and therefore, the type of mindset that one wishing to counter such attacks must adopt.

a. Consider the following C program:

```
begin
print (*begin print (); end.*);
end
```

What do you think the program was intended to do? Does it work?

b. Answer the same questions for the following program:

```
char [] = {'0', ' ', '}', ';', 'm', 'a', 'i', 'n',
'(', ')', '{', and so on... 't', ')', '0'};
main ()
{
int I;
printf(*char t[] = (*);
for (i=0; t[i]!=0; i=i+1)
printf("%d, ", t[i]);
printf("%s", t);
}
```