Analysis and Detection the Computer Script Worms

By

Assis. Prof. Dr. Wesam Samir Bhaya
University of Babylon
Department of Information Network
College of Information Technology
Babylon, Iraq

Assis. Lect. Fairouz Mushtak Jaafar
University of Kufa
Department of Computer Science
College of Education
Najaf, Iraq.

Abstract

A Script worm is defined as a self-replicating and self-containing malicious scripting program that can copy itself to remote computer in network and execute automatically.

In our work, we surveyed different types of script worms according to their spreading method like Drive, Email, Network and Chat script worms, and analyzed the characteristics of their source codes during the target finding and propagation phases of a worm's life cycle. From analyzing process, we extracted general objects and methods that are mostly used by different types of script worms and established the malicious common fixed patterns from them. Analyzed source code and extracted malicious patterns enable us to identify the behavior of worms that could be helpful in developing a broad spectrum of A*nti-Script worm* system software to detect and remove script worms.

When we implemented the proposed Anti-Script worm on the large number of the mentioned above script worms, we saw that this system could detect and remove nearly all the known and unknown script worms. Furthermore, the proposed system could distinguish between the malicious scripts and the benign script files that contained the suspicious *objects* and *methods* by checking all commands in every script files.

1. Introduction

Worm program was apparently first described by John Brunner in 1975 in his classic science fiction novel *The Shockwave Rider*. He called these programs tapeworms that lived ''inside'' the computers and spread themselves to other machines. In 1979-1981, researchers at Xerox PARC built and experimented with worm programs[7].

The computer Worm is a program that is designed to copy itself from one computer to another, leveraging some network ways: email, TCP/IP, etc.. The main difference between a virus and a worm is that a worm does not need a host document. In other words, a worm does not need to attach itself to another program. In that sense, a worm is self-contained[4][10].

Computer Worms can be classified based on the transport and launch mechanism. The

transport mechanism are Email, i.e. using MS Outlook to send itself as an email attachment or the worm may have an SMTP module implemented to create and send mails on its own. A worm may use arbitrary protocols like IRC (Internet Relay Chat) or TCP/IP. If a worm does not "require user interaction in order to gain control of a system", it's called "self-launching worm". An "user-launched" worm must be started by an user, e.g. double-clicking on an infected email attachment. If a worm uses both mechanism, it's called "hybrid-launch worm". Also computer worm can be classified by compiler. Most worms are compiled with Visual Basic, C and Delphi[6].

Many of the worms which managed to cause significant outbreaks use more than one propagation method as well as more than one infection technique[8].

Script worms are not really a new class of worms, but has only quite recently evolved into a major threat, they can travel from machine to machine, preferably over Email. Script worms are written in scripting language as pure text and thus easily readable for everybody. Since computers cannot understand text instructions directly, the text first has to be translated from text to machine code. This procedure is called "interpretation", and is performed by separate programs on the computer. For example, Visual Basic Script

(VBS) and Java Script (JS) are interpreted by the program WSCRIPT.EXE, and old DOS batch language (BAT) is interpreted by COMMAND.COM[3].

The base scripting languages do not have the ability to affect the system on their own, they have no persistent storage. In order to reach out and manipulate the operating system they must make use of objects and their methods. Most of these objects were provided for just this purpose. The most objects that are used by script worms are the *Scripting.FileSystemObject* (FSO), *WScript.Shell*, *WScript.Network*, and *Outlook.Application* objects. The first three exist on all machines running the Windows Scripting host. The *Outlook.Application* object is only present if the target machine has Outlook installed. This reduces the number of viable targets, but facilitates easy spreading. By using the above objects and their method, worms can quickly got accustomed to the Windows operating system and started to send themselves via e-mail, IRC, and other network functions[5].

2. Related Works

This work discusses the malicious codes related to script worms programs like Visual Basic Script and JavaScript worms. Different computer security related researches define the malicious codes of script worm in different words, but the core remains the same.

Prabhat Kumar Singh(2002) attempts in his thesis to identify the various functional organs of a class of malicious code called virus and worm. He provided a detailed physiology for a class of programs after doing an anatomy on them. Also he observed that the different viruses and worms, spaced by the time of their occurrence in the wild, had very similar source code. Sometimes, parts of source code in a virus or worm seemed to have been copied from old viruses or worms respectively[9].

To combat script worm threat, Mark Kennedy(2000) injected an intelligent layer, a script firewall, to determine which scripts are allowed to execute. This layer can be customized to the individual or organization to balance the business requirements against the security implications, then explored the difficulties of building a script behavior blocking system and examined how effective such a system is against today's malicious threats[5].

Robert Fried(2000) focuses on Internet and electronic mail based viruses and worms and how they function. Moreover, the languages in which these viruses are written and distributed, specifically Microsoft's Visual Basic Scripting (VBS) and Hyper Text Markup Language (HTML), and examples of each are discussed in detail. In addition, mechanisms for defense and prevention against such viruses and worms are also evaluated[2].

Matt Bishop(2000) reviewed the structure and organization of the LoveLetter script worm, he presented an analysis of the characteristics that a system must have to be affected, and how the worm affects those systems. He then extrapolated to discuss prevention and confinement of the danger, and also he concluded with some suggestions about protecting systems[1].

3. Problem definition

Our goal is to propose a detection system against script worm, it is essential that the proposal detection system not only detect the known script worm, but it also detect the new and unknown script worm. Our problem definition is based on :

3.1. Study the Codes of Script Worms (Analysis Phase)

After studying and analyzing a large number of script worms codes(nearly eighty samples)[13], we deduced that most of them use the following ***objects and methods***:

- **Scripting.FileSystemObject**
- **Wscript.Shell**
- **Wscript.Network**
- **Outlook.Application**

According to our analysis to the ***objects and methods*** that are used by script worms , we can classify the behavior of script worms, as follows:

I. Drive Script Worm: The drive worm spreads and replicates itself by the external memory such as flash using the autoplay feature and *autorun.inf* file[12], according to our analyzing we see that most of Drive script worms use the objects and methods that are listed in figure(1), these extracted objects and methods represent the malicious pattern for the Drive script worm.

Figure (1): Objects and Methods which are Extracted from All Known Drive Script Worm

A- Drive Script Worm Objects:
- *Scripting.FileSystemObject*
- *Wscript.Shell*

B- Drive Script Worm Methods:
- *GetSpecialFolder*
- *GetFile* (Wscript.ScriptFullName)
- *Drives*
- *DriveType*
- *CreateTextFile*(create scriptworm in system)
- *Copyfile* (copy script worm to system)
- *GetFile*
- *CreateTextFile* (in different drives)
- *GetFile*
- *CreateTextFile*(create *autorun.inf* file in different drives)
- *Run - regwrite*

II. Email Script Worm: Email worms spread via infected email messages. The worm may be in the form of an attachment or the email may contain a link to an infected website[8]. The worm code activates when the infected attachment (with .js or .vbs extension) is opened or when the link to the infected file is opened. Figure (2) shows the most objects and methods which are used by Email worms to achieve its mission. We can use these objects and methods as a generic malicious pattern to detect and remove most Email script worms.

A- Email Worm Script Objects:
- *Scripting.FileSystemObject*
- *Wscript.Shell*
- *Outlook.Application*

B- Email Script Worm Methods
- *CreateTextFile*(create the worm in system)
- *Copy* (copy script worm to system)
- *GetSpecialFolder*
- *GetFile* (Wscript.ScriptFullName)
- *RegWrite*
- *GetNameSpace(MAPI)*
- *AddressList*
- *AddressEntries*
- *CreateItem*
- *To - Subject - Body -Send*
- *Attachements.Add(script worm)*

Figure (2): Objects and Methods that are Used by Most Email Script Worms

III. Network Script Worm: Network Worms propagate via computer networks. The distinguishing feature of this type of worm is that it does not require user action in order to spread[5]. This type of worm usually searches for critical vulnerabilities in software running on networked computers. In order to infect the computers on the network, the worm sends a specially crafted network packet (called an exploit) and as a result the worm code (or part of the worm code) penetrates the victim computer and activates.[8]

Most Network script worm do that using the following objects and methods that are illustrated in figure(3), we can consider these objects and methods as a malicious pattern to detect the most Network script worms.

A- Network Script Worm Objects:
- *Scripting.FileSystemObject*
- *Wscript.Network*

B- Network Script Worm Methods
- *CreateTextFile* (in C drive)
- *GetFile*
- *CopyFile(vbs or js file to the startup fold)*
- *MapNetworkDrive*
- *EnumNetworkDrives*
- *RemoveNetworkDrive*

Figure (3): Objects and Methods that are Used by most Network Script Worms

IV. IRC(Chat) Script Worm: Chat worms perhaps the least popular type of worm spreads via Internet Relay Chat. Like Email worms, Chat Worms have two ways of spreading via IRC channels. The first involves sending an URL which leads to a copy of the worm. The second technique is to send an infected file to an IRC channel user[11]. Two ways use objects and methods that are listed in figure (4).

A- Chat Script Worm Objects:
- *Scripting.FileSystemObject*
- *Wscript.Shell*

B- Chat Script Worm Methods:
- *CreateTextFile(script worm)*
- *FileExit (irc , mirc , pirc or script.ini)*
- *CreateTextFile (script.ini)*
- *GetFile - RegWrite - Run*

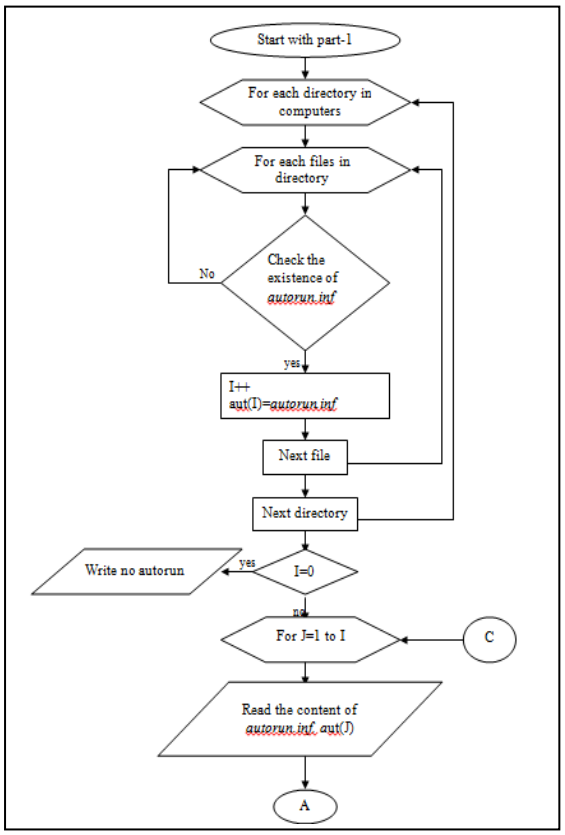
Figure (4): Objects and Methods that are Used by Most Chat Script Worms

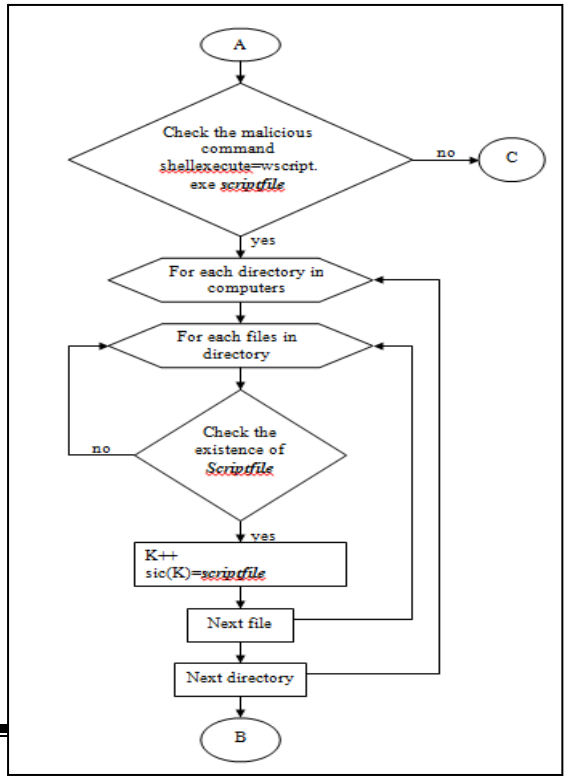
3.2. The Proposed Anti-Script Worm System (Detection Phase)

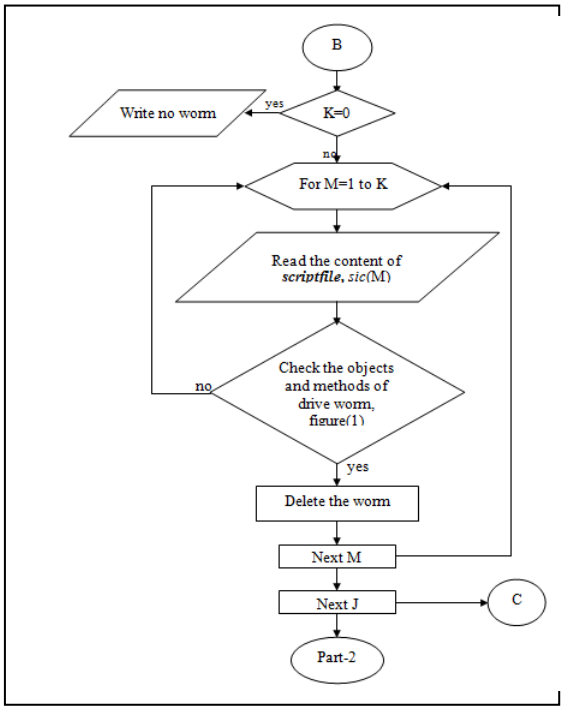
The structure of Anti-Script worm program is divided into two parts, the following detection algorithm demonstrate the two parts in details as follows:

Part-1: Detection and Removing the Drive Script Worm

It is well known that the Drive script worm is activated by *autorun.inf* file unlike the other worms, therefore we need special treatment to detect it. Below the flowchart in figure(5) (and equivalent Pseudocode) explain the part-1 of detecting algorithm.





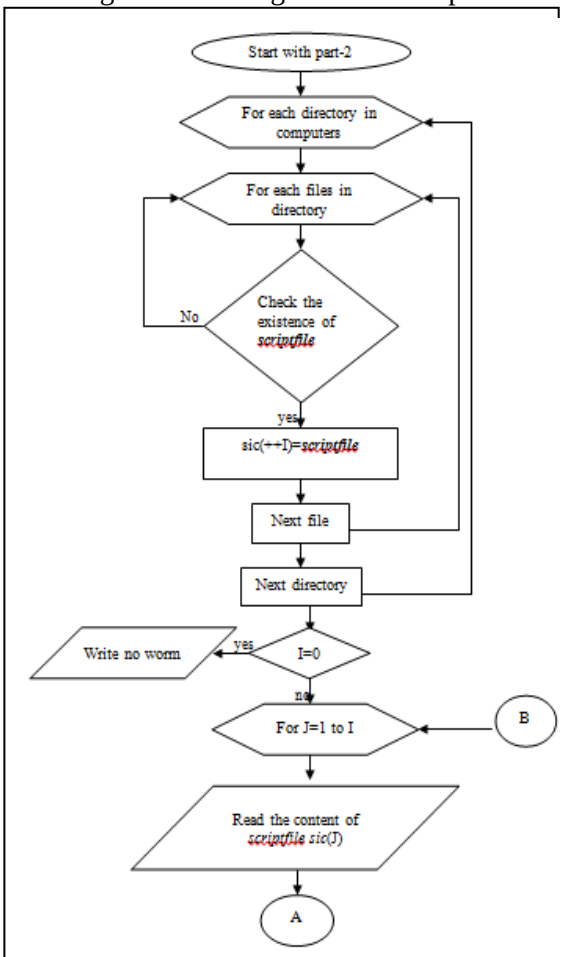


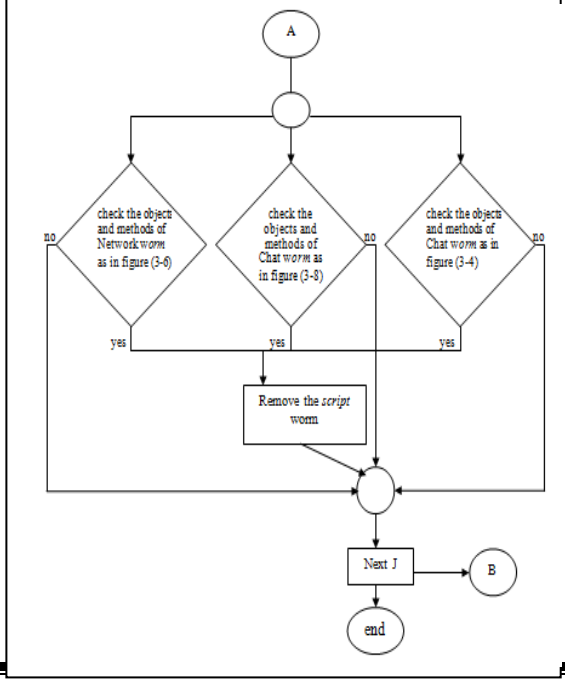
Figure(5): Flow Chart of Detecting and Removing Drive Script Worms

for each directory in computer drives
  for each file in directory
    if the ***autorun.inf*** file is found ***then***
      aut(I++)=***autorun.inf***
  next file
next directory
if I=0 ***then*** **write** "no autorun file"
else
for J=1 to I
  read aut(J), the contents of each founded ***autorun.inf*** file
  if these content have the malicious command "shellexecute=wscript.exe ***scriptfile***" or "open= wscript.exe ***then***
    for each directory in computer drives
      for each file in directory
        if scriptfile is found ***then***
          sic(K++)=***scriptfile***
        else
      next file
    next directory
        if K=0 ***then***
        write "no drive worm"
        else
    for M=1 to K
      read sic(M), the contents of each founded ***scriptfile***
      if sic(M)=objects and methods of drive worm as in figure(1)
        then **delete** the worm
    next M
next J

Part-2: Detection and Removing the Other Script Worms Types

Below the flowchart in figure(6) (and equivalent Pseudocode) explain part-2 of the detecting and removing the other script worms





Figure(6): Flow Chart of Detecting and Removing Email, Network and Chat Script

```
for each directory in computer drives,
    for each file in directory
        'check the existence of the scriptfile
        if this file is found then
            sic(I++)= scriptfile
    next file
next directory
if I=0 then write "no script worm"
else
for J=1 to I
    'check the contents of scriptfile
  Select case sic(J)
      Case objects and methods of Email
            worm as in figure (2)
          write "delete the Email worm"
      Case objects and methods of Network
            worm as in figure (3)
          write "delete the Network worm"
      Case objects and methods of Chat worm
            as in figure (4)
          write "delete the Chat worm"
      Case else
          write "no script worm"
    End select
Next J
```

4. Conclusions

In our paper, when studying the script worm source code, we saw a frequent observation that the different script worms had very similar source code. Parts of source code in a worm seemed to have been copied from old worms. Those worms that had remarkably different source codes displayed identical program behavior. A conclusion from this observation is that detecting most script worms by studying previous worm behaviors is possible, that mean it is possible to detect unknown worms and future ones.

5. Results

The proposed system could distinguish between the malicious scripts and the benign ones that contained the suspicious *objects* and *methods* by checking all commands in every script files in computer drives to examine the absence or presence the malicious patterns which are extracted from analysis phase, if these patterns were found then the checked script file was a worm otherwise was a benign script file. In other words the proposed system could detect only the script file that contain the entire malicious pattern and ignore the script file that contain part of malicious pattern

When compared our modeling results with results of other traditional detection systems such as Norton and Rising antivirus, we concluded that this proposed system could detect and remove the whole script worm file, unlike the others that could detect the script worm and remove only the malicious command from it.

6. Future Steps

During our study, we found some new research areas. Below are some suggestions for further work that extend this thesis:

- Extends the script worm code analysis to manipulate the encrypted and polymorphic script worm.
- Extends the proposed Anti-Script worm system to evaluate the misclassification, for example the false alarm rate or false positive, in order to better evaluate the detection capabilities
- This thesis can be used as the basis model for binary worm classification, this type of worm is embedded itself in other file with encoded form.

7. References

[1] Bishop M., **2000**, Analysis of the ILOVEYOU Worm, *University of California; Department of Computer Science*; Davis CA 95616-8562.

[2] Fried R., **2000**, Beware of Your Inbox!,*www.crime-scene-investigator.net/BewareInbox.pdf.*

[3] Jaquet C.; Ness Y., **2002**, The Norman Book on Computer Viruses, *Norman* ASA,.

[4] Kak A., **2010**, Lecture Notes on "Computer and Network Security", *Purdue University*.

[5] Kennedy M., **2000 ,** Script-Based Mobile Threats, *Symantec AntiVirus Research Center*,

[6] Link R., **2003,** Server-based Virus-protection On Unix/Linux, *Diploma Thesis in University of Applied Sciences Furtwangen*; Faculty of Computer

Science ; Computer Networking; Germany.
[7] Nazario J. , **2004**, Defense and Detection Strategies against Internet Worms, *Artech House; INC; Computer Security Series*.
[8] Securelist: **2010,** Viruses and worms, *http://www.securelist.com/en/ threats/detect/viruses-and-worms*,
[9] Singh P., **2002**, A Physiological Decomposition of Virus and Worm Programs, *Thesis in University of Louisiana at Lafayette*; College of Science; India,.
[10] Szor P., **2005**, The Art of Computer Virus Research and Defense, *Symantec Corp*..
[11] Szor P., **2001**, Virus Analysis 1, *Symantec Corporation*.
[12] Tahir R.; Hamid Z.; H. Tahir**,2008**, Analysis of AutoPlay Feature via the USB Flash Drives, *Proceedings of the World Congress on Engineering*; Vol. I; London; U.K.
[13] VX heavens, **2010**, Computer virus collection*/Worm.VBS.Autorun,http://vx.net lux.org/vl.php?dir=Worm.VBS.Autoru.*

بحث مقدم من قبل

الأستاذ المساعد الدكتور وسام سمير بهية

جامعة بابل / كلية تكنولوجيا المعلومات/ قسم شبكات المعلومات

العراق / بابل

المدرس المساعد فيروز مشتاق جعفر

جامعة الكوفة / كلية التربية للبنات/ قسم الحاسبات

العراق / النجف الأشرف

المستخلص

تعرف الدودة النصية (Script worm) بأنها برنامج لدودة خبيثة مكتوبة بلغة نصية (script) وهي برنامج مستقل بحد ذاته يستنسخ نفسه ذاتيا الى الحاسبات الاخرى في الشبكة.

في عملنا هذا قمنا بعرض انواع مختلفة من الديدان النصية بالاعتماد على طرق انتشارها مثل الدودة التي تنتقل عن طريق البريد الالكتروني او شبكة الحاسوب او المحادثة او عن طريق الفلاش، ثم قمنا بتحليل خصائص ايعازات برامج هذه الانواع عند اكتشافها في الهدف وخلال دورة حياتها. ومن عملية التحليل تمكنا من استخلاص مجموعة من الدوال والوظائف العامة والتي تعتبر اكثر استخداما من قبل أنواع مختلفة من الديدان النصية، ثم تكوين قوالب عامة وثابتة وخبيثة. إن تحليل الايعازات واستخلاص القوالب الخبيثة مكننا من تمييز سلوك الدودة النصية والذي أفادنا في تطوير نظام واسع مضاد يقوم بكشف وإزالة هذه الدودة.

عند تطبيق النظام المقترح على عدد كبير من الديدان النصية المذكورة أعلاه لاحظنا أن النظام قادر على كشف وازالة كل الديدان النصية المعروفة وغير المعروفة. اضافة الى ان النظام قادر على التميز بين الديدان الخبيثة والملفات النصية الحميدة والتي تحوي على نفس الدوال والوظائف المشكوك فيها عن طريق اجراء اختبار وفحص كل الايعازات في كل ملف نصي.