Scan of the Month: Scan 32

Clarence Ford
Kevin Habel
Jeremy Thompson
Dwight Pierce
Robert Nicholson Jr.
William Rettman
Matt Schoenecker
Rasheed Graham
Instructor: Jeremy Hansen, CISSP

October 1, 2004

Contents

1 Introduction

This month's challenge was to analyze a home-made malware binary. This home-made malware binary was created to reinforce the value of reverse engineering. More details about this challenge are available on the Honey net website.

2 Analysis Objectives

Objectives of this analysis are to:

- **2.1** Identify and provide an overview of the binary, including the fundamental pieces of information that would help in identifying the same specimen.
- **2.2** Identify and explain the purpose of the binary
- **2.3** Identify and explain the different features of the binary. What are its capabilities
- **2.4** Identify and explain the binary communication methods. Develop a Snort signature to detect this type of malware being as generic as possible, so others similar specimens could be detected, but avoiding at the same time a high false positive rate signature
- **2.5** Identify and explain any techniques in the binary that protect it from being analyzed or reversed engineered
- **2.6** Categorize this type of malware (virus, worm…) and justify your reasoning
- **2.7** Identify another tool that has demonstrated similar functionality in the past
- **2.8** Suggest detection and protection methods to fight against the threat introduced by this binary
- **2.9** Bonus Questions:
  - **2.9.1** Is it possible to interrogate the binary about the person who developed this tool?
    - **2.9.1.1** In what circumstances and under which conditions?
  - **2.9.2** What advancements in tools with similar purposes can we expect in the near future?

3 Plan of analysis for Scan of the Month Project "Scan32"

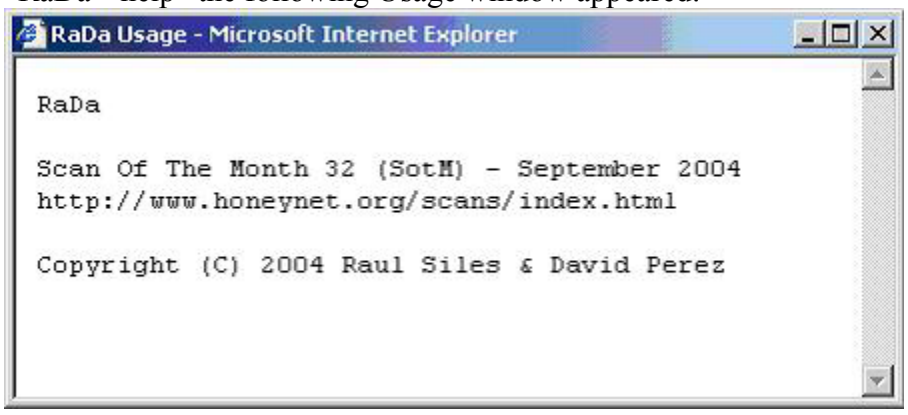
- **3.1** Run strings commands to extract the available printable text from the binary.
- **3.2** Run the binary through a disassembler (pedasm) to view source code.
- **3.3** Run binary in a debugger Debuggy for line by line analysis and monitor the memory.
- **3.4** Install MD5Sum to take an MD5 of all the files on System prior to running the binary, and then take another hash for comparison.
- **3.5** Run Diskmon to monitor the hard drives activity.
- **3.6** Run FileMon to monitor and display file system activity on a system in real-time.
- **3.7** Run Regmon and Regprot to monitor the registry for attempts to modify it
- **3.8** Run packet sniffer (Ethereal) to monitor traffic
- **3.9** Monitor logs from Apache and firewall

4 Activity Analysis

- **4.1 Here is a sample of the output from the strings commands**

CompanyName
Malware
ProductName
RaDa
FileVersion
1.00
ProductVersion
1.00
InternalName
RaDa
OriginalFilename
RaDa
VarFileInfo
Translation
!This program is the binary of SotM 32..
Rich
:)D
Form1
Module1
v.%
Command_instal
833q
You c
var
ot play/g fun
ny securit
ch@e
usag
exit
conf
Label
s,@68'<
,,_
L0nkn`r
WTC5T
P2u
dx"k)k7
EVENT
_SINK_R
-gEq
FunSion
KERNEL32.DLL
MSVBVM60.DLL
LoadLibraryA
GetProcAddress
ExitProcess

From this output we were determined then the binary contains Visual Basic 6 coding. The reference to "usag" indicated that a help function may exist. But typing "RaDa --help" the following Usage window appeared:



This gave us the names of the authors of the binary. The reference to Form1 and Label led us to look for GUI options. By typing "RaDa --gui" the following window appeared:

Pressing "Uninstall" gives a run-time error and quits. "Install" copies itself to C:\rada\bin\ and creates c:\rada\tmp as well as adding itself to run in the registry. "Show config" and "Show usage" both pop up the Usage window. The "GO!" button attempts to connect to 10.10.10.10:80, and exit just exits.

4.2 Here is a sample of the output from the disassembler (pedasm):

```
Label40fd6c ::
      xor ecx , ecx
      sub eax , 03h
      jb Label40fd80
      shl eax , 08h
      mov  al , byte ptr [esi]
      inc esi
      xor eax , 0FFFFFFFFh
      je Label40fdf2
      mov ebp , eax
```

The use of the xor function suggests to us that some encryption may be occurring.

4.3 The debugger program revealed some of the DLL that were being used, which indicated possible functions that the binary was attempting to perform.

4.3.1 RPCRT4.DLL – contains Remote Procedure Calls API which allows for network and Internet communication.

4.3.2 SCRRUN.DLL – contains libraries that allow reading and writing of scripts and text files

4.4 The MD5Sum showed that the following files were modified:

4.4.1 MSIMGSIZ.DAT

4.4.2 ---\Cookies\index.dat

4.4.3 ---\History.IE5\index.dat

4.4.4 ---\content.IE5\index.dat

This suggest that the binary was attempting to remove "evidence" of the activities of the worm

4.5 Diskmon did not provide any new information

4.6 The registry monitor showed us that the binary:

4.6.1 Created the HKLM\Software\Microsoft\Windows\CurrentVersion\Run\RaDa.

4.6.2 Accessed the HKLM\SOFTWARE\Microsoft\Cryptography\RNG key

4.6.3 And access several other keys

From this we determined that the binary added itself to run on startup and was using the Random Number Generator related to the Cryptography services

4.7 The packet sniffer reported the binary was attempting the following

4.7.1 to connect to a web site at 10.10.10.10

4.7.2 reverse DNS lookup

```
4.7.2.1     15.10.10.10.in-addr.arpa    [itself]
4.7.2.2     10.10.10.10.in-addr.arpa    [webserver]
4.7.2.3     100.1.168.192.in-addr.arpa [gateway])
```

Ethereal also shows that the binary broadcasts NetBIOS Name Query packets (port 137). It also broadcasts NetBIOS datagrams (SMB packets) which holds information about the Windows network (port 138).

4.8 Apache Logs reported the following :

```
====================
= Apache access_log =
====================

10.10.10.15 - - [30/Sep/2004:01:14:18 -0500] "GET
/RaDa/RaDa_commands.html HTTP/1.1" 404 304    (404 means not found)


This shows what binary attempts to access at 10.10.10.10 (Source IP =
10.10.10.15)

====================
= Apache error_log =
====================

[Thu Sep 30 01:14:18 2004] [error] [client 10.10.10.15] File does not
exist: /var/www/htdocs/RaDa/RaDa_commands.html


========================================
= Results of http request to 10.10.10.10 =
========================================

GET /RaDa/RaDa_commands.html HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: 10.10.10.10
Connection: Keep-Alive

HTTP/1.1 404 Not Found
Date: Thu, 30 Sep 2004 06:14:18 GMT
Server: Apache/1.3.31 (Unix) PHP/4.3.7
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

124
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /RaDa/RaDa_commands.html was not found on this
server.<P>
<HR>
<ADDRESS>Apache/1.3.31 Server at whirled.dyndns.org Port 80</ADDRESS>
</BODY></HTML>
```

5 Results

Identify and provide an overview of the binary, including the fundamental pieces of information that would help in identifying the same specimen

- Copies itself to C:\rada\bin\ and creates c:\rada\tmp as well as adding itself to run in the registry.
- Attempts to connect to http://10.10.10.10/RaDa/RaDa_commands.html
- It attempts to remove traces of internet activity

Identify and explain the purpose of the binary

- We believe the purpose of this binary is to propagate and to spread and maybe open a backdoor via the commands that it looks for on the website.

Identify and explain the different features of the binary. What are its capabilities?

- We believe that it can communicate via the Windows network and the Internet
- We believe that it has the ability to write scripts and text files
- We believe that it has the ability to self-propagate

Identify and explain the binary communication methods. Develop a Snort signature to detect this type of malware being as generic as possible, so others similar specimens could be detected, but avoiding at the same time a high false positive rate signature

- Alert tcp any any -> any 80 (content: "RaDa_commands.html")

Identify and explain any techniques in the binary that protect it from being analyzed or reversed engineered

- The use of the xor function suggests to us that some encryption may be occurring.

Categorize this type of malware (virus, worm…) and justify your reasoning

- We believe that this binary is a worm. We believe this because we believe that it has the ability to self-propagate across the Internet and network shares.

Identify another tool that has demonstrated similar functionality in the past

- Worm.Win32 Opasoft
    - The worm installs itself and sets to auto-run
    - Opasoft scans subnets for port 137 in order to find victim computers
    - Opasoft sends, via port 139

Suggest detection and protection methods to fight against the threat introduced by this binary

- The use of any combination of the following systems would be helpful in fighting the threat introduced by this binary
    - The use of a personal firewall
    - Intrusion Detection Systems
    - Registry monitors
    - Intrusion Prevention Systems

Bonus Questions:
Is it possible to interrogate the binary about the person who developed this tool, in what circumstances and under which conditions?

- Yes, in by running the binary in GUI mode or accessing the help function the binary identifies the authors. (Raul Siles & David Perez)