

Introduction

Developed by Jean-Pierre Lesueur, the DarkComet RAT (Remote Administration Tool) is a powerful and controversial piece of software designed for remote access and control of computers. It gained notoriety for its capabilities in allowing administrators to remotely manipulate systems over the internet. Originally intended for legitimate remote administration tasks, DarkComet RAT has unfortunately been utilized for malicious purposes due to its extensive feature set due to its key features which include remote access control, keystroke logging, webcam and microphone access, file management, and more; making it a versatile tool for both legitimate administrators and malicious actors alike. Its stealth capabilities allow it to operate discreetly on compromised systems, often evading detection by antivirus software.

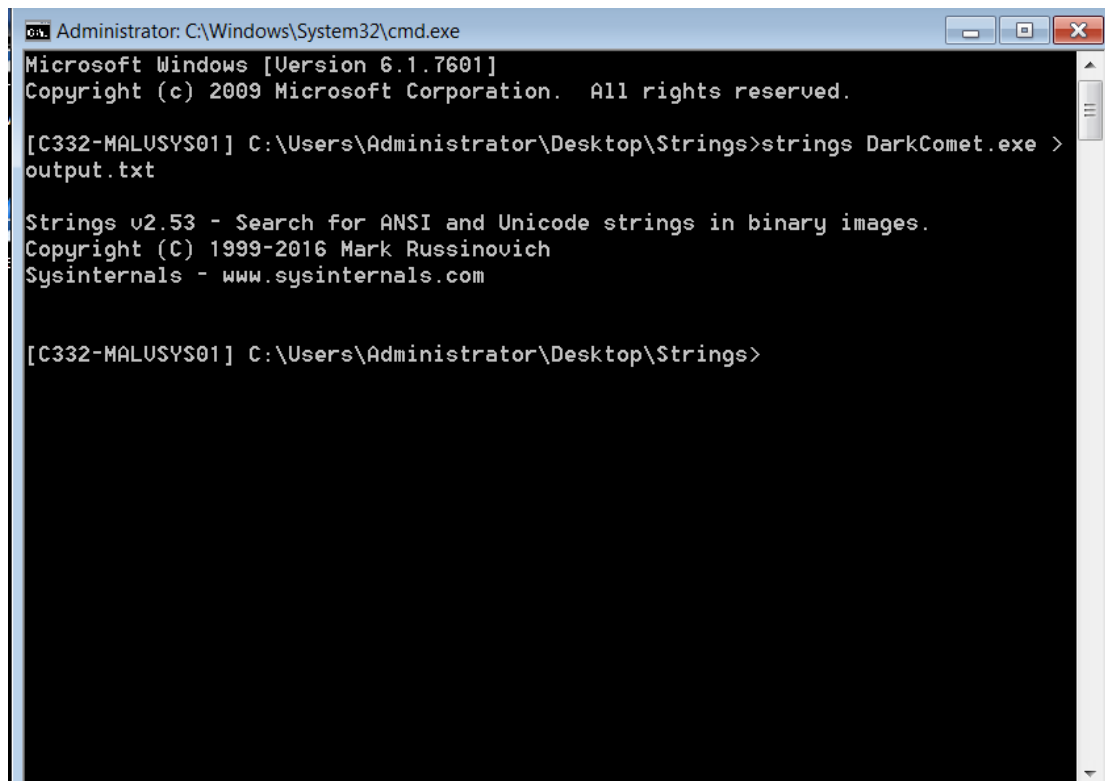
Due to its association with cybercrime and unauthorized access, DarkComet RAT has been the subject of scrutiny and legal actions in various jurisdictions. Despite this, its functionalities continue to attract attention from both cybersecurity professionals and those looking to exploit vulnerabilities for illicit purposes.

The purpose of this report is to conduct a detailed analysis of the DarkComet RAT, elucidating its behavior, functionalities, and impact on affected systems. By systematically examining its code, execution patterns, and network interactions, this analysis aims to provide insights into the malware structure as well as its detection.

Malware Information

File Name	DarkComet.exe
File Type	Portable Executable 32
File Info	Borland Delphi 4.0
File Description	A remote administration tool from the cosmos
File Version	4.2.0.28
MD5	D761F3AA64064A706A521BA14D0F8741
SHA1	AB7382BCFDF494D0327FCCCE9C884592BCC1ADEB

Malware Breakdown and Analysis



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

[C332-MALUSYS01] C:\Users\Administrator\Desktop\Strings>strings DarkComet.exe >
output.txt

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

[C332-MALUSYS01] C:\Users\Administrator\Desktop\Strings>
```

Figure 0: Extracting ASCII and Unicode strings from DarkComet binary

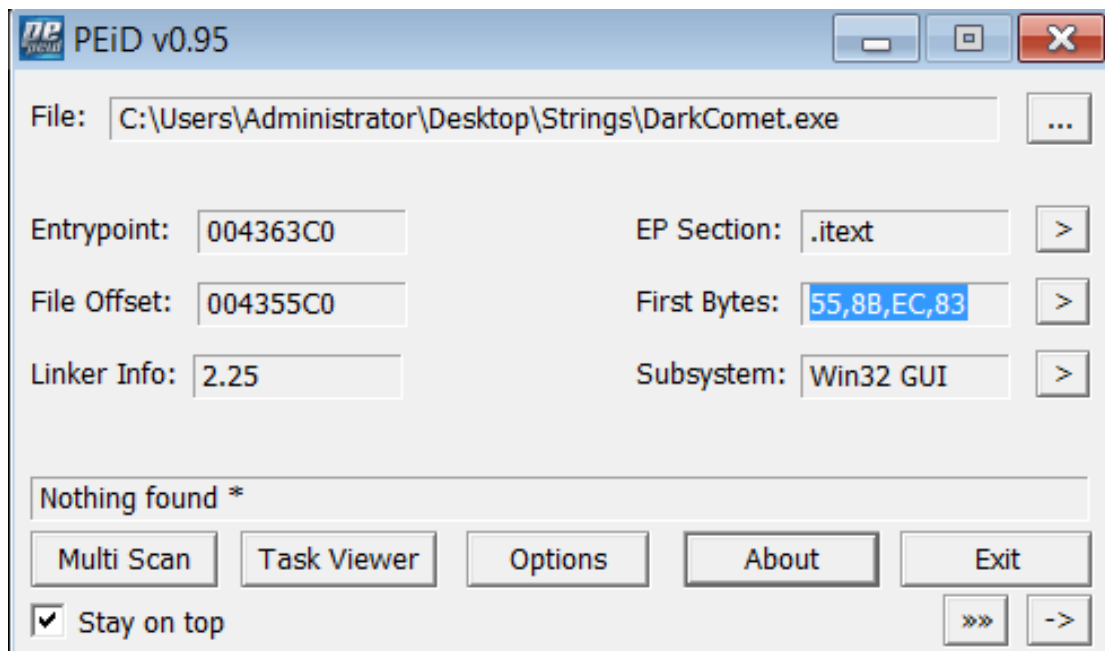


Figure 1: DarkComet binary passed through PEiD

The first bytes of the file are “ 55, 8B,EC,83 ” - which is interesting enough to keep in consideration

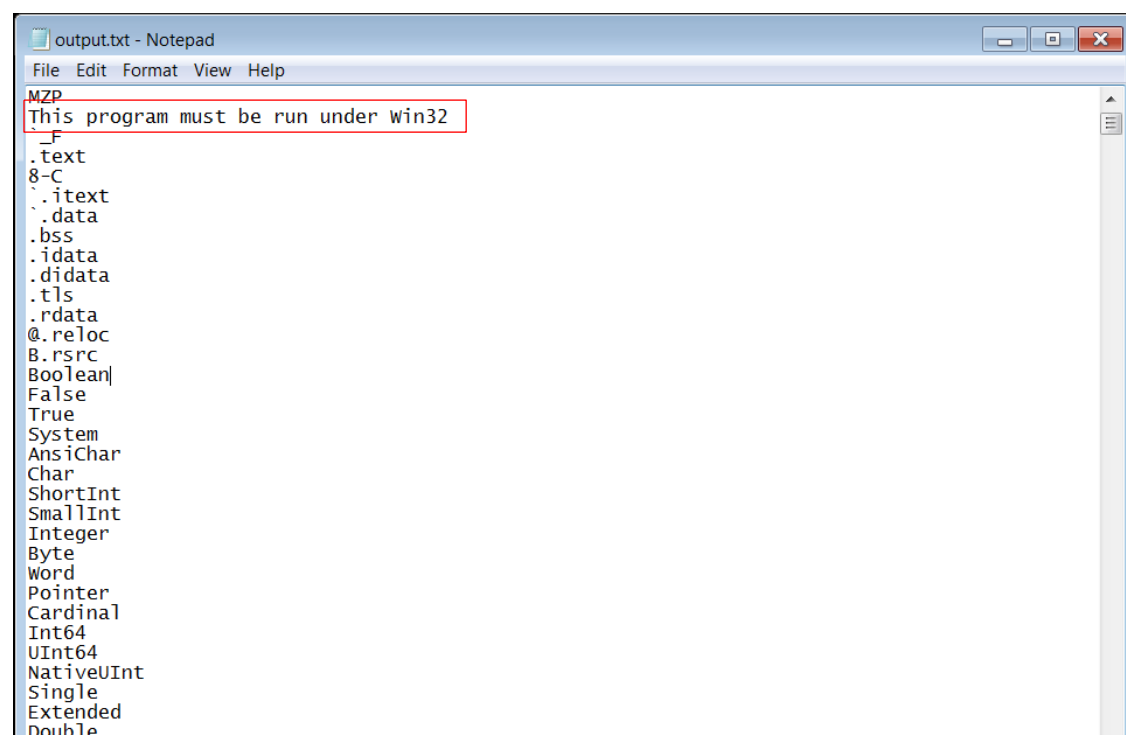
The hexadecimal values, which often represent machine code instructions or data. Below is a breakdown of each byte:

55: This hexadecimal byte corresponds to the machine code instruction PUSH EBP. In x86 assembly language, PUSH EBP is used to push the value of the EBP register onto the stack. This is commonly used in function prologues to save the base pointer before setting up a new stack frame.

8B: This byte is part of a larger instruction. In x86 assembly, 8B is the opcode prefix for various MOV (move) instructions. The specific function depends on the following byte.

EC: This hexadecimal byte corresponds to the machine code instruction PUSH ESP. In x86 assembly, PUSH ESP pushes the value of the ESP register (the stack pointer) onto the stack. This is used in some scenarios to save the stack pointer before modifying it.

83: This byte is also part of a larger instruction set. In x86 assembly, 83 is the opcode prefix for various arithmetic and logic operations using immediate values. The specific operation depends on the following byte.



```
output.txt - Notepad
File Edit Format View Help
MZP
This program must be run under win32
F
.text
8-C
.itext
.data
.bss
.idata
.didata
.tls
.rdata
@.reloc
B.rsrc
Boolean
False
True
System
AnsiChar
Char
ShortInt
SmallInt
Integer
Byte
Word
Pointer
Cardinal
Int64
UInt64
NativeUInt
Single
Extended
Double
```

Figure 2: Initial information dereived from DarkComet binary

In conclusion, these bytes are part of the machine code representation of instructions that execute on the x86 architecture, which confirms with the output.txt that DarkComet RAT primarily focuses on 32-bit versions of Microsoft Windows Operating System, including Windows 95, 98, NT, 2000, XP, Vista, 7, and 8.

DarkComet Unique Identifier

MUTEX (Mutual Exclusion Object) is a synchronization primitive used in programming to ensure that only one thread or process accesses a resource at a time. DarkComet creates a MUTEX to ensure that only one instance of itself is running on the system at any point of time in order to prevent duplication of its processes and execute smoothly without any interference.

The MUTEX created by the DarkComet is unique due to it being presented in the following format:

DC_Mutex-XXXXXXX

The X is represented by any Alphanumeric character and the total count is equal to 7 . The GUI has a feature to randomize it

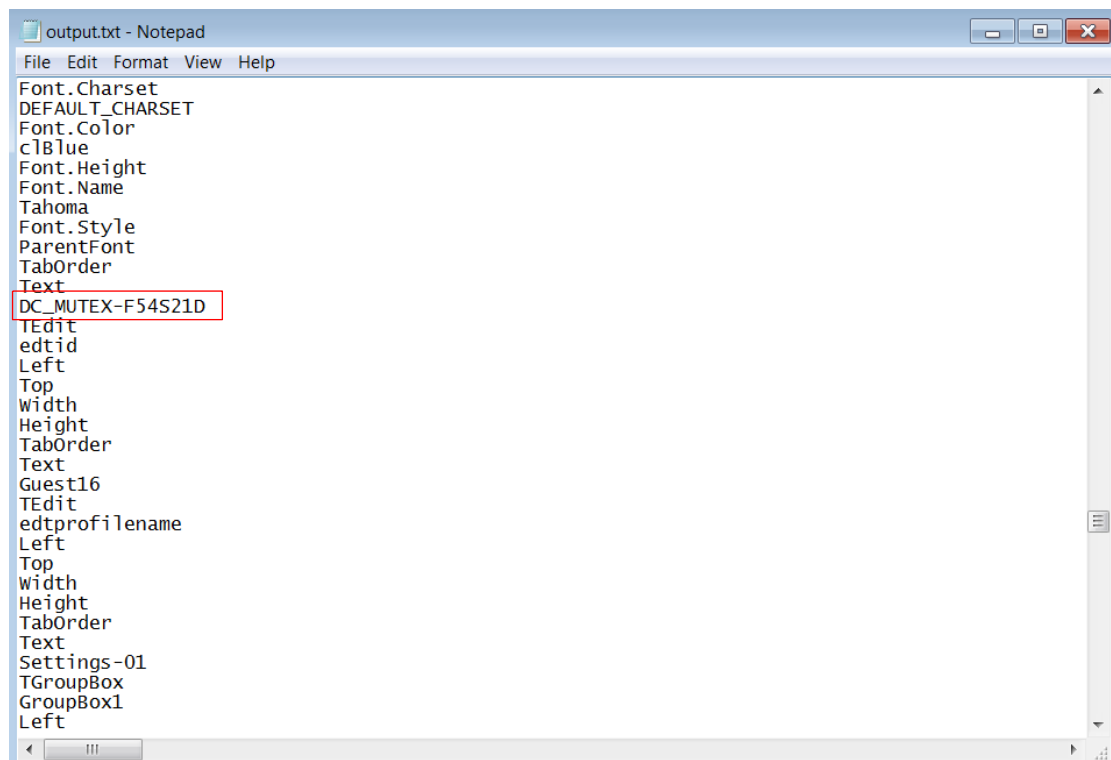


Figure 4: DarkComet Mutex for particular binary

DarkComet has many other unique Strings and features. I have compiled a comprehensive set of YARA rules designed to identify instances of DarkComet RAT based on unique identifiers extracted from known samples. These rules meticulously target specific strings, byte sequences, and behavioral patterns associated with DarkComet RAT's operations, ensuring robust detection capabilities with the aim to proactively identify and mitigate the threat posed by DarkComet RAT, bolstering the defence against potential abuse of malicious remote access Trojan if it ever appears once more in the wild.

Malware Functions and DLLs used

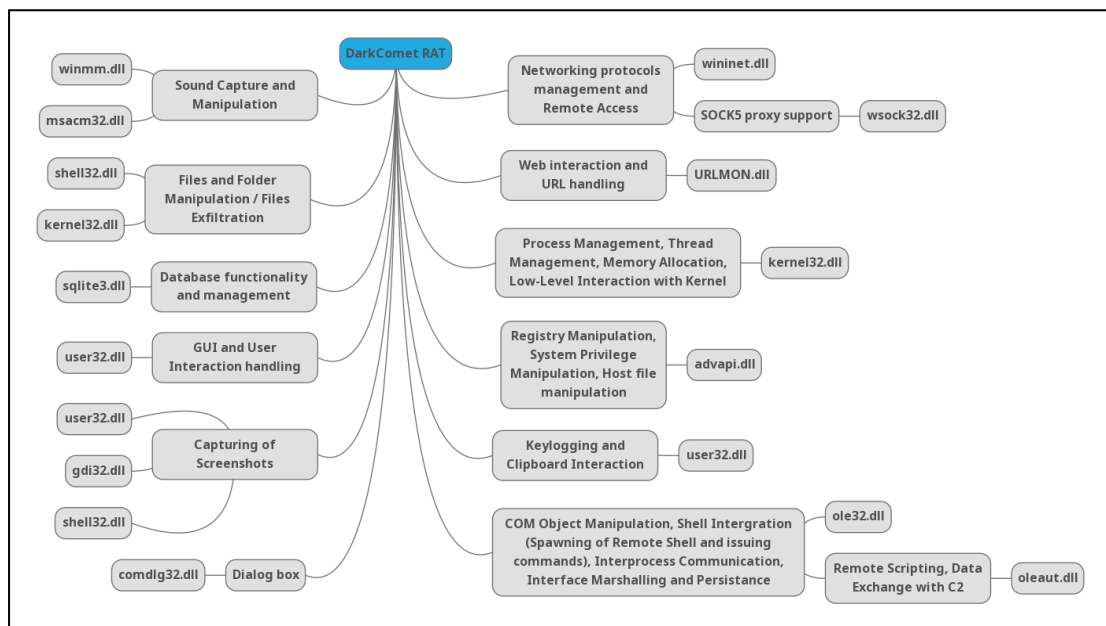


Figure 3: A breakdown of DarkComet functions and the DLLs used