



Investigate Windows Management Instrumentation (WMI) Attacks in Windows Operating Systems

**A dissertation submitted for the Degree of Master of
Science in Information Security**

K A M Maduranga
University of Colombo School of Computing
2016



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Students Name: K A Milan Maduranga

Signature:

Date:

This is to certify that this thesis is based on the work of Mr. K A Milan Maduranga under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. Ajantha Athukorala

Signature:

Date:

Abstract

Windows operating system has one powerful technology that has remained consistent since Windows 95 that is Windows Management Instrumentation (WMI) and this existing on all Windows operating systems since windows 95. WMI is contained very powerful set of tools used to manage Windows systems both locally and remotely.

In information security world a major threat is that of intruders which may maliciously try to access the data or services on the remote system using the inbuilt tools in operating systems. WMI has been well known and heavily utilized by system administrators since its launch and WMI has become very popular tool between attackers because its ability to perform system reconnaissance, code execution, lateral movement, anti-virus and virtual machine detection, persistence, and data theft.

WMI use Distributed Component Object Model(DCOM) as its default protocol for communication over the network. DCOM establishes an initial connection over TCP port 135. Subsequent data is then exchanged over a randomly selected TCP port using Distributed Computing Environment/Remote Procedure Call (DCE/RPC) protocol. There are several mitigations that may prevent WMI attacks from occurring. These are close the port 135 or disable WMI in the hosts. But problem was then network administrators also cannot use this service for their tasks. Also port 135 not only used for the WMI communication. This port used by many other Microsoft services.so it's impossible to close this port with the systems.

Distributed Computing Environment/Remote Procedure Call (DCE/RPC) header include field called operation number. With this operation number value can identify that traffic contain WMI activity or not. Also command executed by remote user can find from the Stub data field inside the DCE/RPC header.

An Intrusion Detection System is a method used for monitoring the network and protecting it from the intruder. A better solution is to use a device or software that can immediately detect and stop an attack. Intrusion Prevention System performs this function. This research has implemented WMI Intrusion Detection System and WMI Intrusion Protection System for protect networks from WMI based attacks. For that implementation used knowledge gaining from this research.

Acknowledgement

I am grateful to Dr. Ajantha Athukorala, for her guidance as my supervisor. He always encouraged me to pursue research and give correct guidance for the project.

I would like to thank lecture panel including Dr. Kasun De Zoysa, Dr.D.N. Ranasinghe ,Dr. Hakim Usoof and Mr.Kenneth for giving me valuable comments in improving this project.

I would specially thank my family for the support they gave me during the time I have to spend for this research. Without their support, I would not even think about furthering my study and this research work.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION	1-2
1.1 Project Introduction	1
1.2 Project Scope	2
1.3 Desertion Structure	2
CHAPTER 2-BACKGROUND, LITERATURE REVIEW & RELATED WORK	3-20
2.1 Interact with Windows Management Instrumentation	
2.1.1 PowerShell	3
2.1.2 Windows Management Instrumentation Command line	3
2.1.3 Windows Remote Management Command Line Tool	4
2.2 Windows Management Instrumentation Attacks	4
2.2.1 Reconnaissance	4
2.2.2 Anti-Virus Detection	6
2.2.3 VM Detection	6
2.2.4 Code Execution and Lateral Movement	6
2.3 Windows Management Instrumentation Attack History	7
2.4 Common Information Model	9
2.5 Windows Management Instrumentation	9
2.6 Distributed Component Object Model (DCOM)	9
2.7 Key Microsoft Server Products that used Network port 135	10
2.8 Distributed Computing Environment/Remote Procedure Call protocol	12
2.9 Software tools	12
2.10 Intrusion Detection System(IDS)	13
2.11 Intrusion Protection System(IPS)	14
2.12 IPS and IDS Categories	14
2.13 How Remote WMI works	15
2.14 Related Works	17
2.14.1 WMI Classes and Namespaces with Different OS	17
2.14.2 Deep analyses of WMI attacks	18
CHAPTER 3 – DESIGN	21-32
3.1 Security Implementations of Networks	21
3.2 Design WMI Intrusion Detection System(IDS)	22
3.3 Design WMI Intrusion Protection System(IPS)	24
CHAPTER 4 – IMPLEMENTATION	26-32
4.1 WMI Intrusion Detection System(IDS) with snort	26
4.2 Implementation of WMI based IDS using python	28
4.2 Implementation of WMI based IPS using python	30
CHAPTER 5 - RESULTS ANALYSIS AND DISCUSSION	33-36
CONCLUSION	37-38
REFERENCE	39

LIST OF FIGURES

Figure 01: Windows Management Instrumentation Command line	03
Figure 02: Windows Remote Management Command Line Tool	04
Figure 03-07: Remote WMI Query Execution on remote host	05
Figure 08: Remote WMI Execution for get virus guard information	06
Figure 09: Remote WMI Execution for detect Virtual machine	06
Figure 10-11: Remote WMI Execution for start process on remote PC	07
Figure 12: WMI DCOM initial establishment over TCP port 135	10
Figure 13: DCE /RPC Header Structure	12
Figure 14: How IPS works	14
Figure 15: How IDS works	14
Figure 16: WMI communication between two devices	15
Figure 17: WMI communication between two devices via Wireshark	16
Figure 18: WMI Header with Wireshark	16
Figure 19: WMI Classes and Namespaces	17
Figure 20: WMI pattern analyse with CSV	18
Figure 21: WMI pattern analyse with Wireshark	19
Figure 22: DCE RPC Operation Number with Wireshark	19
Figure 23: DCE RPC stub data with Wireshark	20
Figure 24: Promiscuous Monitoring in Network	22
Figure 25: IDS Design Flowchart	23
Figure 26: IP header total Length field	23
Figure 27: WMI Intrusion Protection System flow chart	24
Figure 28: WMI Intrusion Detection System(IDS) with snort	26
Figure 29: WMI Intrusion Detection System(IDS) snort output	27
Figure 30-31: python WMI Detection tool output	29
Figure 32: log file structure	30
Figure 33: TCP header	31
Figure 34: connection reset with WMI IDS	32
Figure 35-36: WMI Firewall Interface	32
Figure 37: log file generated by WMI IDS tool	34
Figure 38: log file analysis with spunk	35
Figure 39: log file analysis with spunk	35

LIST OF TABLES

Table 01- Windows Management Instrumentation Attack History	07
Table 02- Key Microsoft Server Products that used Network port 135	11
Table 03- WMI Classes and Namespaces with operating systems	17

LIST OF ABBREVIATIONS

WMI	Windows Management Instrumentation
DCE	Distributed Computing Environment
RPC	Remote Procedure Call
PDU	Protocol data unit
OS	Operating System
TCP	Transmission Control Protocol

CHAPTER 1 - INTRODUCTION

1.1 Project Introduction:

In the Windows operating system, one powerful technology that has remained consistent since Windows 95 is Windows Management Instrumentation (WMI). WMI Present on all Windows operating systems, WMI is contained of a powerful set of tools used to manage Windows systems both locally and remotely.

WMI has been well known and utilized heavily by system administrators since its Beginning, WMI became popular in the security community when it was found to be used by Stuxnet 3. Since then, WMI has been gaining popularity amongst attackers because of its ability to perform system reconnaissance, anti-virus and virtual machine (VM) detection, code execution, lateral movement, persistence, and data theft. As attackers gradually utilize WMI, it is important for security defenders, incident responders, and forensic analysts to have knowledge of WMI and to know how they can apply it to their advantage.

WMI is a default service installed on Windows XP and Server 2003 and all Microsoft Operating System. So can freely write WMI scripts or applications to automate administrative tasks on Windows-based systems. WMI acts as a means to acquire information on how an OS operates. It helps administrators to get extract information about all aspects of an OS.

WMI is a useful tool for system administration and computer management. However, the ability and features of WMI are also potential tools for threat distribution. So WMI tool can be more useful or harmful tool according to the incident.

There are many ways to interact with WMI and PowerShell can use easily when interacting with WMI remotely. So Attackers are using built in components of Windows systems that are extremely powerful.

Using WMI can gather information about a system's services, processor, disk, and all objects information. Also WMI can automate hardware and software data collection and it can be used to automate malicious activities. WMI supports scripting and it can allow malicious scripts to be embedded in and carried out by the normal service. Attackers Use WMI for different type of attacks types including Reconnaissance, Lateral movement, Privilege escalation, Establishing a foothold, Persistence, Data theft etc.

Windows PowerShell is most powerful way to interact with WMI and that allows for a multitude of result formatting options. Attackers can use WMI to connect to remote systems with PowerShell, to modify the registry, access event logs, and most important and execute commands.

According to the **security week**¹ website they tell that last 6 months of the 2014 an increase in attackers using PowerShell and WMI for post-compromise activity.

1.2 Project Scope

WMI is a useful tool for system administration and computer management. However, the ability and features of WMI are also potential tools for threat distribution. In this research deeply observe WMI activity and find security solution for network protect from WMI attack. Normally by disabling the WMI service and blocking TCP port 135 can mitigate from these attack. Then problem was network administrators also cannot use WMI for their tasks. And cannot close port 135 from firewall because its uses more Microsoft services other than WMI. so need sound security solution for mitigate from the WMI attacks. Finally need to implement Intrusion detection/protection system for detect malicious WMI activity.

1.3 Deseration Stucture

in this deseration cahpter one discuss about the basic introduction about the research and cover aims and objectives of the project. This chapter entirely given the idea about the structure of the project and some important information about Windows Management Instrumentation. From chapter two discuss the applied the fundamental theories, standards, concepts and previous research work related to WMI and DCERPC. Hence it is important to show how those theories, standards and concepts are linked to the project in order to achieve the aims and objectives. Also in this chapter include information about some software tool that directly related to this project implementation. Chapter three outlines the step by step design process, System was improved by starting fundamental model. Also discuss about the Security Implementations of Networks using WMI based IDS and IPS Systems. Chapter four outlines the step by step Implementation process IPS and IDS. Chapter five discuss about results analysis and future works.

CHAPTER 2 - BACKGROUND, LITERATURE REVIEW & RELATED WORK

2.1 Interact with Windows Management Instrumentation

To interact with WMI can used different tools. Interacting with WMI Microsoft and third party vendors provide a wealth of client tools that allow you to interact with WMI.

2.1.1 PowerShell

PowerShell is an extremely powerful scripting language that contains a wealth of functionality for interacting with WMI. As of PowerShell version 3 the following cmdlets are available for interacting with WMI:

Get-WmiObject	Get-CimAssociatedInstance	Get-CimClass
Get-CimInstance	Get-CimSession	
Set-WmiInstance	Invoke-WmiMethod	
Invoke-CimMethod	New-CimInstance	
New-CimSession	New-CimSessionOption	
Register-CimIndicationEvent	Register-WmiEvent	
Remove-CimInstance	Remove-WmiObject	
Remove-CimSession		

The WMI and CIM cmdlets offer similar functionality; however, CIM cmdlets were introduced in PowerShell version 3 and offer some additional flexibility over WMI cmdlets.

Not all systems will have PowerShell v3+ installed, however. PowerShell v2 is installed by default on Windows 7. As such, it is viewed as the least common denominator by attackers.

2.1.2 Windows Management Instrumentation Command line

```
C:\Users\milan m>wmic
wmic:root\cli>
```

Figure 01: Windows Management Instrumentation Command line

wmic.exe is a powerful command line utility for interacting with WMI. It has a large amount of convenient default aliases for WMI objects but you can also perform more complicated queries.

wmic.exe can also execute WMI methods and is used commonly by attackers to perform lateral movement by calling the Win32_Process Create method. One of the limitations of wmic.exe is that you cannot call methods that accept embedded WMI objects.

If PowerShell is not available though, it is good enough for performing reconnaissance and basic method invocation. wmic.exe is still used commonly by pentesters and attackers.

2.1.3 Windows Remote Management Command Line Tool

```
C:\Users\milan m>winrm
Windows Remote Management Command Line Tool

Windows Remote Management (WinRM) is the Microsoft implementation of
the WS-Management protocol which provides a secure way to communicate
with local and remote computers using web services.

Usage:
  winrm OPERATION RESOURCE_URI [-SWITCH:VALUE [-SWITCH:VALUE] ...]
  [@{KEY=VALUE[;KEY=VALUE]...}]
```

Figure 02: Windows Remote Management Command Line Tool

winrm.exe can be used to enumerate WMI object instances, invoke methods, and create and remove object instances on local and remote machines running the WinRM service. winrm.exe can also be used to configure WinRM settings. The ideal method of interacting with WMI over WinRM is PowerShell. WinRM has superseded DCOM as the recommended remote management protocol for Windows.

WinRM was also built to support WMI or more generically, CIM operations over the network. By default, the WinRM service listens on TCP port. WinRM settings are easily configurable using GPO, winrm.exe, or the PowerShell.

2.2 Windows Management Instrumentation Attacks

WMI is an extremely powerful tool for attackers across many phases of the attack lifecycle. There is a wealth of WMI objects, methods, and events that can be extremely powerful for performing anything from reconnaissance, AV/VM detection, code execution, lateral movement, covert data storage, to persistence. It is even possible to build a pure WMI backdoor that doesn't introduce a single file to disk.

There are many advantages of using WMI to an attacker: It is installed and running by default on all Windows operating systems (after Windows 98). For code execution, it offers a stealthier alternative to running psexec. Permanent WMI event subscriptions run as SYSTEM. Defenders are generally unaware of WMI as a multi-purpose attack vector. Nearly every operating system action is capable of triggering a WMI event. Other than storage in the WMI repository, no payloads touch disk. The following is a exhaustive list of how WMI can be used to perform the various stages of an attack.

2.2.1 Reconnaissance

Reconnaissance is the checking out a situation before taking an action and this is the One of the first steps taken my most malware and pentesters. WMI has a huge number of classes that can help an attacker get a feel for the environment they're targeting. These are some of the more common reconnaissance tasks carried out by attackers and the respective WMI objects that can be queried. Figure 3 to 7 show that how attackers get information of remote host using PowerShell .

OS related information

```
Win32_OperatingSystem
Win32_ComputerSystem
```

Class Win32_BIOS

Disk volume list: Win32_Volume

```
PS C:\> Get-WmiObject Win32_Volume -ComputerName 192.168.10.10 -Credential %computername%\administrator

__GENUS           : 2
__CLASS           : Win32_Volume
__SUPERCLASS     : CIM_StorageVolume
__DYNASTY         : CIM_ManagedSystemElement
__RELPATH        : Win32_Volume.DeviceID="\\\\.\\Volume{01c04603-d4b2-11e3-9219-806e6f6e6963}\\"
__PROPERTY_COUNT : 44
__DERIVATION     : (CIM_StorageVolume, CIM_StorageExtent, CIM_LogicalDevice, CIM_LogicalElement...)
__SERVER         : WIN-0J0HKI5SDQM
```

Figure 03: Remote WMI Execution for Win32_Volume

List service: Class Win32_Service

```
PS C:\> Get-WmiObject Win32_Service -ComputerName 192.168.10.10 -Credential %computername%\administrator

ExitCode : 0
Name     : AeLookupSvc
ProcessId : 0
StartMode : Manual
State    : Stopped
Status   : OK
```

Figure 04: Remote WMI Execution for Win32_Service

Logs: Win32_NtLogEvent

```
PS C:\> Get-WmiObject Win32_NtLogEvent -ComputerName 192.168.10.10 -Credential %computername%\administrator

Category           : 0
CategoryString     : 
EventCode          : 1001
EventIdentifier    : 1001
TypeEvent          : 
EventStringsArray : (0) AeLookupSvc, Transition, Net, Local, ...
```

Figure 05: Remote WMI Execution for Win32_NtLogEvent

Logged On User: Win32_LoggedOnUser

```
PS C:\> Get-WmiObject Win32_LoggedOnUser -ComputerName 192.168.10.10 -Credential %computername%\administrator

__GENUS           : 2
__CLASS           : Win32_LoggedOnUser
__SUPERCLASS     : CIM_Dependency
__DYNASTY         : CIM_Dependency
__RELPATH        : Win32_LoggedOnUser.Antecedent="\\\\.\\root\\cimv2:Win32_Account.Domain=\"WIN-0J0HKI5SDQM\",Name=\"SYSTEM\"",Dependent="\\\\.\\root\\cimv2:Win32_LogonSession.LogonId=\"999\""
```

Figure 06: Remote WMI Execution for Win32_LoggedOnUser

Share : Win32_share

```
PS C:\> Get-WmiObject Win32_share -ComputerName 192.168.10.10 -Credential %computername%\administrator

Name           Path           Description
----           -
ADMIN$         C:\Windows    Remote Admin
C$             C:\           Default share
IPC$           C:\           Remote IPC
```

Figure 07: Remote WMI Execution for Win32_share

2.2.2 Anti-Virus Detection

Installed AV products will typically register themselves in WMI via the AntiVirusProduct class contained within either the root\SecurityCenter or root\SecurityCenter2 namespaces depending upon the OS version.

Below example show how to get information about Virus Guard That installed on PC.

Ex: Get-WmiObject -Namespace root\SecurityCenter2 -Class AntiVirusProduct

```
PS C:\> Get-WmiObject -Namespace root\SecurityCenter2 -Class AntiVirusProduct -C
omputerName 192.168.10.10 -Credential %computername%\administrator

__GENUS                : 2
__CLASS                 : AntiVirusProduct
__SUPERCLASS           :
__DYNASTY               : AntiVirusProduct
__RELPATH               : AntiVirusProduct,instanceGuid="{4D41356F-32AD-7C42-C
820-63775EE4F413}"
__PROPERTY_COUNT       : 5
__DERIVATION            : {}
__SERVER                : WIN-0J0HKI5SDQM
__NAMESPACE            : ROOT\SecurityCenter2
__PATH                  : \\WIN-0J0HKI5SDQM\ROOT\SecurityCenter2:AntiVirusProd
uct,instanceGuid="{4D41356F-32AD-7C42-C820-63775EE4F
413}"
displayName             : AVG AntiVirus Free Edition
```

Figure 08: Remote WMI Execution for get virus guard information

2.2.3 VM Detection

Also detection of Virtual Machine can be performed. For example, if physical memory is less than 2GB or if there is only a single processor core, it is likely you're running in a VM.

Figure 07 show that PowerShell script to detect remote host is virtual or not

```
1 $VMDetected=$False
2 $Arguments= @{}
3 $Class = 'Win32_ComputerSystem'
4 $Filter = 'NumberOfLogicalProcessors < 2 OR TotalPhysicalMemory <
5 2147483648'
6 }
7 if (Get-WmiObject@Arguments) { $VMDetected=$True }
```

Figure 09:Remote WMI Execution for detect Virtual machine

2.2.4 Code Execution and Lateral Movement

Lateral movement consists of techniques that enable to access and control remote systems on a network and run tools on remote systems

The Win32_Process class contains a static method named Create that can start a process locally or remotely. This is the WMI equivalent of running psexec.exe only without unnecessary forensic artifacts like the creation of a service.

The following example (figure 10,11) demonstrates executing a process on a remote machine:

```
PS C:\> Invoke-WmiMethod -Class Win32_Process -Name Create -ArgumentList 'notepad.exe' -ComputerName 192.168.10.10 -Credential %computername%\administrator

__GENUS           : 2
__CLASS           : __PARAMETERS
__SUPERCLASS     : 
__DYNASTY        : __PARAMETERS
__RELPATH        : 
__PROPERTY_COUNT : 2
__DERIVATION     : {}
__SERVER         : 
__NAMESPACE     : 
__PATH           : 
ProcessId        : 2864
ReturnValue       : 0
```

Figure 10: Remote WMI Execution for start process on remote PC
In victim's PC

Image Name	User Name	CPU	Memory (...)	Description
notepad.exe	Administrator	00	736 K	Notepad
SearchIndexe...	SYSTEM	00	4,840 K	Microsoft ...
services.exe	SYSTEM	00	3,592 K	Services ...
smss.exe	SYSTEM	00	208 K	Windows ...

Figure 11: start process on remote PC

2.3 Windows Management Instrumentation Attack History

This section discusses about the WMI attack history and also discusses about the tools that use to the WMI activity.

Name of the attack	Description
Deep Panda	WMI for lateral movement
APT29	WMI to steal credentials and execute backdoors at a future time
Lazarus Group malware	to start itself on a target system during lateral movement
Stealth Falcon malware	gathers system information via Windows Management Instrumentation
The DustySky	to extract information about the operating system and whether an anti-virus is active
BlackEnergy	WMI to gather victim host details

Table 01- Windows Management Instrumentation Attack History

The Deep Panda

Deep Panda is a suspected Chinese threat group known to target many industries, including government, defence, financial, and telecommunications. The intrusion into healthcare company Anthem has been attributed to Deep Panda. This group is also known as Shell Crew, WebMasters, KungFu Kittens, and Pink Panther. Deep Panda also appears to be known as Black Vine based on the attribution of both group names to the Anthem intrusion.

Usage of WMI:

PowerShell - Deep Panda has used PowerShell scripts to download and execute programs in memory, without writing to disk.

Windows Management Instrumentation - The Deep Panda group is known to utilize WMI for lateral movement.

Windows Admin Shares - Deep Panda uses net.exe to connect to network shares using "net use" commands with compromised credentials.

APT29

APT29 is threat group that has been attributed to the Russian government and has operated since at least 2008.¹² This group reportedly compromised the Democratic National Committee starting in the summer of 2015.

Usage of WMI:

Windows Management Instrumentation Event Subscription - APT29 has used WMI event filters to establish persistence.

Windows Management Instrumentation - APT29 used WMI to steal credentials and execute backdoors at a future time.

Lazarus Group

Lazarus Group is a threat group that has been active since at least 2009 and was reportedly responsible for the November 2014 destructive wiper attack against Sony Pictures Entertainment. It was responsible for a campaign known as Operation Blockbuster. Malware used by Lazarus Group correlates to other reported campaigns, including Operation Flame, Operation IMission, Operation Troy, DarkSeoul, and Ten Days of Rain.

Usage of WMI:

Windows Management Instrumentation - uses the Windows Management Instrumentation Command-line application wmic to start itself on a target system during lateral movement.

Stealth Falcon

Stealth Falcon is a threat group that has conducted targeted spyware attacks against Emirati journalists, activists, and dissidents since at least 2012. Circumstantial evidence suggests there could be a link between this group and the United Arab Emirates (UAE) government, but that has not been confirmed

Usage of WMI:

Windows Management Instrumentation - Stealth Falcon malware gathers system information via Windows Management Instrumentation (WMI).

PowerShell - Stealth Falcon malware uses PowerShell commands to perform various functions, including gathering system information via WMI and executing commands from its C2 server.

System Owner/User Discovery - Stealth Falcon malware gathers the registered user and primary owner name via WMI

Software: DustySky, NeD Worm

DustySky is multi-stage malware written in .NET that has been used by Molerats since May 2015

Usage of WMI:

Windows Management Instrumentation - The DustySky dropper uses Windows Management Instrumentation to extract information about the operating system and whether an anti-virus is active.

Software: BlackEnergy, Black Energy

BlackEnergy is a malware toolkit that has been used by both criminal and APT actors. It dates back to at least 2007 and was originally designed to create botnets for use in conducting Distributed Denial of Service (DDoS) attacks, but its use has evolved to support various plug-ins. It is well known for being used during the confrontation between Georgia and Russia in 2008, as well as in targeting Ukrainian institutions. Variants include BlackEnergy 2 and BlackEnergy

Usage of WMI:

Windows Management Instrumentation - A BlackEnergy 2 plug-in uses WMI to gather victim host details

2.4 Common Information Model

CIM: Common Information Model (CIM) is the Distributed Management Task Force (DMTF) standard [DSP0004] for describing the structure and behavior of managed resources such as storage, network, or software components.

2.5 Windows Management Instrumentation

WMI: Windows Management Instrumentation (WMI) is a CIM server that implements the CIM standard on Windows. WMI is primarily a system of organized classes that represent management information from the Windows® operating system and other Windows-based hardware and software products. A class is really nothing more than an abstract description of the properties and capabilities some given software or hardware component possesses.

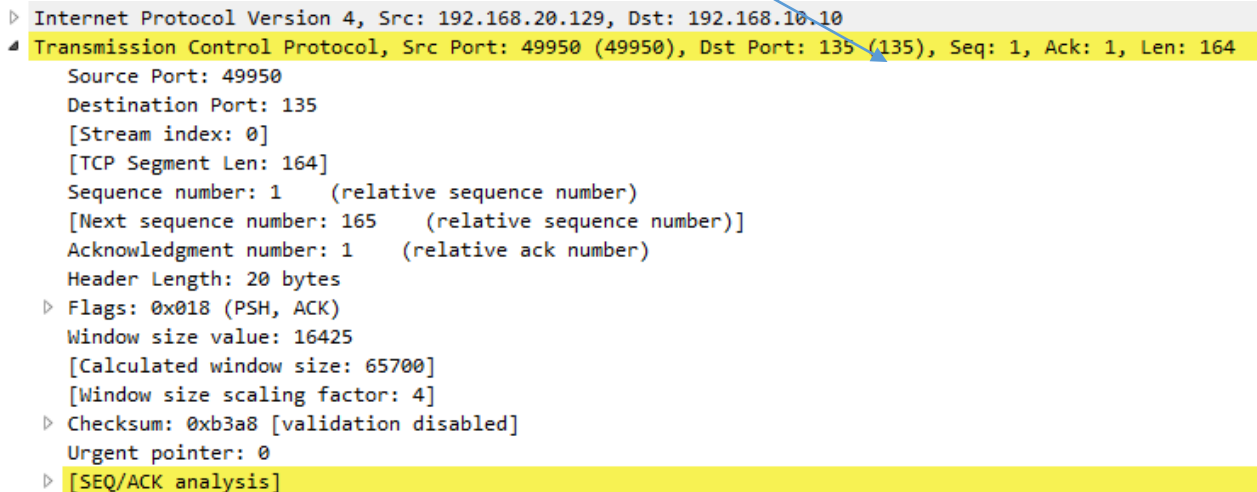
For example, a logical disk class might describe a device that has a serial number, a fixed storage capacity, an amount of available capacity, and so forth. Meanwhile, a class that describes a Windows service might specify that the service has a name, that it can start and stop, its current status, and so on.

2.6 Distributed Component Object Model (DCOM)

DCOM has been the default protocol used by WMI since its initiation. DCOM establishes an initial connection over TCP port 135. Subsequent data is then exchanged over a randomly selected TCP port. This port range can be configured either via dcomcnfg.exe which ultimately modifies the following registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc\Internet – Ports (REG_MULTI_SZ)
All of the built-in WMI cmdlets in PowerShell communicate using DCOM.

DCOM establishes an initial connection over TCP port 135



```

> Internet Protocol Version 4, Src: 192.168.20.129, Dst: 192.168.10.10
* Transmission Control Protocol, Src Port: 49950 (49950), Dst Port: 135 (135), Seq: 1, Ack: 1, Len: 164
  Source Port: 49950
  Destination Port: 135
  [Stream index: 0]
  [TCP Segment Len: 164]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 165 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  Header Length: 20 bytes
  > Flags: 0x018 (PSH, ACK)
  Window size value: 16425
  [Calculated window size: 65700]
  [Window size scaling factor: 4]
  > Checksum: 0xb3a8 [validation disabled]
  Urgent pointer: 0
  > [SEQ/ACK analysis]

```

Figure 12: WMI DCOM initial establishment over TCP port 135

2.7 Key Microsoft Server Products that used Network port 135

1. Certificate Services: Certificate Services is part of the core operating system that enables a business to act as its own certification authority (CA).

2. Cluster Service: The Cluster service controls server cluster operations and manages the cluster database. A cluster is a collection of independent computers that is as easy to use as a single computer. Managers, programmers, and users see the cluster as a single system. The software distributes data among the nodes of the cluster.

3. Distributed File System: The Distributed File System (DFS) service manages logical volumes distributed across a local or wide area network (LAN or WAN) and is required for the Microsoft Active Directory SYSVOL share. DFS is a distributed service that integrates disparate file shares into a single logical namespace.

4. Distributed Link Tracking Server: The Distributed Link Tracking Server system service stores information so that files moved between volumes can be tracked to each volume in the domain.

5. Event Log: This system service logs event messages issued by programs and the Windows operating system. Event Log reports contain information that can be useful in diagnosing problems.

6. Exchange Server: Microsoft Exchange Server includes several system services. When a MAPI client such as Microsoft Outlook connects to an Exchange server, the client first connects to the RPC endpoint mapper (the RPC Locator Service) on TCP port 135.

7. File Replication: The File Replication system service allows files to be automatically copied and maintained simultaneously on multiple servers.

8. Local Security Authority: The Local Security Authority (LSASS) service provides core operating system security mechanisms. It uses random TCP ports assigned through the RPC service for domain controller replication.

9. Message Queuing: The Message Queuing system service is a messaging infrastructure and development tool for creating distributed messaging applications for Windows.

10. Remote Procedure Call: The Microsoft Remote Procedure Call (RPC) system service is a secure inter-process communication (IPC) mechanism that enables data exchange and invocation of functionality residing in a different process. The different process can be on the same computer, on the LAN, or across the globe through a WAN or VPN connection.

11. Remote Storage Notification: The Remote Storage Notification system service notifies users when they read from or write to files that are available only from a secondary storage media. If this service is stopped, notification does not occur.

12. Systems Management Server: Systems Management Server (SMS) 2003 provides a comprehensive solution for change and configuration management for the Microsoft platform, enabling organizations to provide relevant software and updates to users quickly and cost-effectively.

13. Terminal Services Licensing: The Terminal Services Licensing system service installs a license server and provides registered client licenses when connecting to a Terminal Server.

Port	Protocol	Application protocol	System Service Name
135	TCP	RPC	Message Queuing
135	TCP	RPC	Remote Procedure Call
135	TCP	RPC	Exchange Server
135	TCP	RPC	Certificate Services
135	TCP	RPC	Cluster Service
135	TCP	RPC	Distributed File System
135	TCP	RPC	Distributed Link Tracking
135	TCP	RPC	Distributed Transaction Coordinator
135	TCP	RPC	Event Log
135	TCP	RPC	Fax Service
135	TCP	RPC	File Replication
135	TCP	RPC	Local Security Authority
135	TCP	RPC	Remote Storage Notification
135	TCP	RPC	Remote Storage Server
135	TCP	RPC	Systems Management Server 2.0
135	TCP	RPC	Terminal Services Licensing
135	TCP	RPC	Terminal Services Session Directory

Table 02- Key Microsoft Server Products that used Network port 135

2.8 Distributed Computing Environment/Remote Procedure Call (DCE/RPC) protocol

DCE/RPC, short for "Distributed Computing Environment / Remote Procedure Calls", is the remote procedure call system developed for the Distributed Computing Environment. DCE RPC developed by open software foundation (OSF) and Microsoft adopt this as standard. DCERPC Implemented as a true middleware system execute between existing operating system and applications.

Main objective of the DCERPC is to make it possible for a client to access a remote service by simply calling a logical procedure. This system allows programmers to write distributed software as if it were all working on the same computer, without having to worry about the underlying network code.

The following diagram (figure 13) illustrates the point at which the DCE/RPC preprocessor begins processing DCE/RPC traffic for the different transports layer protocol TCP and UDP using port number 135.

Figure 13: DCE /RPC Header Structure



DCE RPC Header Structure

Version	2 bytes
packet type	1 bytes
Packet flags	1 bytes
Data representation	4 bytes
Frag length	2 bytes
Auth length	2 bytes
Call ID	4 bytes
Call hint	4 bytes
opnum	2 bytes
Object UUID	16 bytes

2.9 Software tools

Snort

Snort is an open source software. Snort uses a flexible rule-based language to describe the traffic. This tool records the packet in human readable form. Through protocol analysis, content searching, and various pre-processors Snort detects thousands of worms, vulnerability exploit attempts, port scans, and other suspicious behavior.

Scapy

Implemented python based tool develop with support of the Scapy. scapy is stable python library.this developed by Philippe Biondi as a packet manipulation. Scapy is powerful and flexible, and the possibilities are almost infinite. According Scapy developers Scapy's PCAP processing can be extended to carve out images from HTTP traffic and then perform facial detection on them to determine if there are humans present in the images.For this implementation scapy used to identify the header structure of captured packets.

Kivy

Kivy runs on Linux, Windows, OS X, Android and iOS. You can run the same code on all supported platforms. Kivy is an open source Python library. For this implementation Kivy used to add the GUI interface for the implemented python program.

Wireshark

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level.

Splunk

Splunk is well-known log analyzing application used by industry.

The Transmission Control Protocol

<https://tools.ietf.org/html/rfc793>

2.10 Intrusion Detection System(IDS)

Administrators need to continuously monitor the network and analyze the log files generated by the network devices to protect network from intruders. But that is a time-consuming task and provides a limited view of the attacks being launched against a network. Also in-between the time the logs are analyzed, the attack may have already been successful.

An Intrusion Detection System is a method used for monitoring the network and protecting it from the intruder.

Intrusion Detection Systems (IDSs) were implemented to passively monitor the traffic on a network. IDS-enabled device copies the traffic stream and analyzes the copied traffic rather than the actual forwarded packets.

IDS Working offline, it compares the captured traffic stream with known malicious signatures, like software that checks for viruses. Even if the traffic is monitored and maybe reported, no action is taken on packets by the IDS. This offline IDS implementation is referred to as promiscuous mode. As the IDS is not inline, it has no impact on network performance.

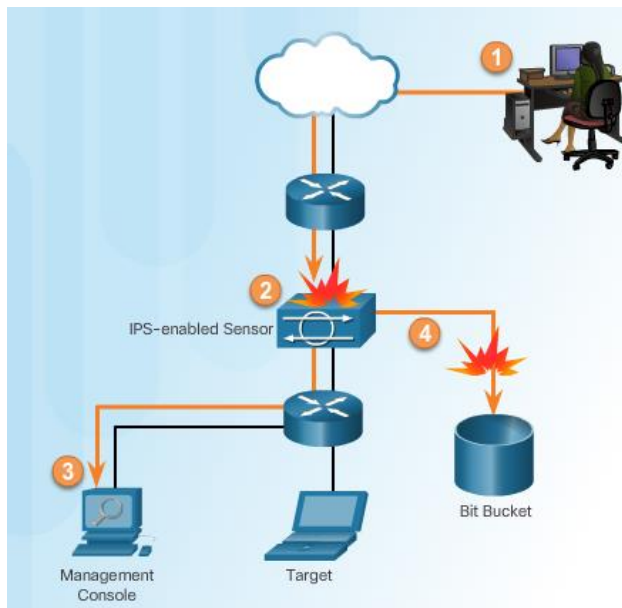


Figure 14: How IPS works

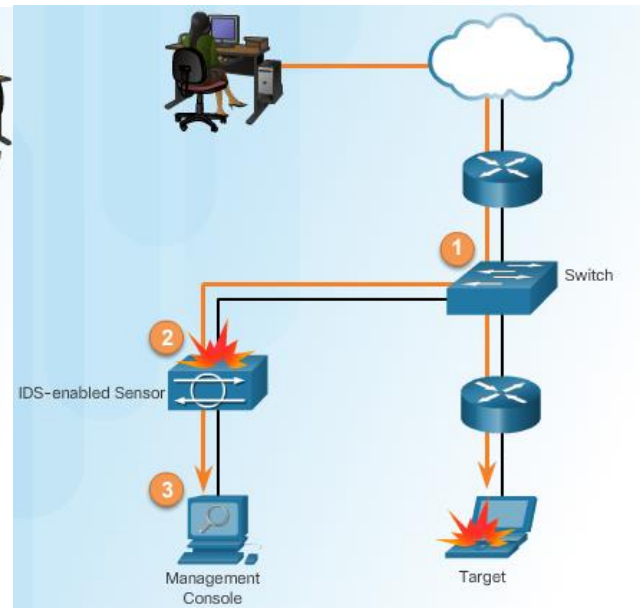


Figure 15: How IDS works

2.11 Intrusion Protection System(IPS)

A better solution is to use a device that can immediately detect and stop an attack (figure 14). A WMI IPS System performs this function.

One of the most powerful actions that an IPS device can perform is to drop packets or prevent an activity from occurring. IPS can be enabled to deny the attacker packets, deny the connection, or deny the specific packet.

Dropping packets enables the device to stop an attack before it has the chance to perform malicious activity. Unlike a traditional IDS device, the IPS device actively forwards packets across two of its interfaces. The analysis engine determines which packets should be forwarded and which packets should be dropped.

2.12 IPS and IDS Categories

Intrusion Detection/Prevention system is classified into two categories: signature based detection systems, anomaly based detection systems

Signature Based /Pattern Based

The network must be able to identify incoming malicious traffic in order to stop it. Fortunately, malicious traffic displays distinct characteristics or “signatures”. A signature is a set of rules that an IDS and an IPS use to detect typical intrusion activity

signature is a pattern that corresponds to a known threat. In signature based detection, observed events are compared against the pre-defined signatures in order to identify possible unwanted traffic. This type of detection technique is very fast and easy to configure.

Signature-based detection is very effective at detecting known threats but largely ineffective at detecting previously unknown threats, threats disguised by the use of evasion techniques, and many variants of known threats. An attacker can slightly modify an attack to render it undetectable by a signature based IDS.

Anomaly Based

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. The profiles are developed by monitoring the characteristics of typical activity over a period of time.

The major benefit of Anomaly based detection technique is that they can be very useful for detecting unwanted traffic that is not specifically known. Also sometimes legitimate traffic can also detect as attack and then that can be disadvantage. For instance, anomaly-based IDS will detect that an Internet protocol (IP) packet is malformed. There are 3 main type IDS available, Host based IDS, Network based IDS and Application based IDS

Network Based IDS (NIDS)

Network based IDS systems collect information from the network itself. The NIDS audits the network attacks while packets moving across the network.

NIDS are deployed on strategic point in network infrastructure. The NIDS can capture and analyse data to detect known attacks by comparing patterns or signatures of the database or detection of illegal activities by scanning traffic for anomalous activity. NIDS are also referred as “packet-sniffers”, Because it captures the packets passing through the of communication medium.

Atomic Signature:

An atomic signature is the simplest type of signature. It consists of a single packet, activity, or event that is examined to determine if it matches a configured signature. If it does, an alarm is triggered, and a signature action is performed. Because these signatures can be matched on a single event, with atomic signatures, the entire inspection can be accomplished in an atomic operation that does not require any knowledge of past or future activities.

2.13 How Remote WMI works

WMI communication in between two device use port 135 and this use special protocol called DCERPC for command execution. Figure 16 demonstrate this process.

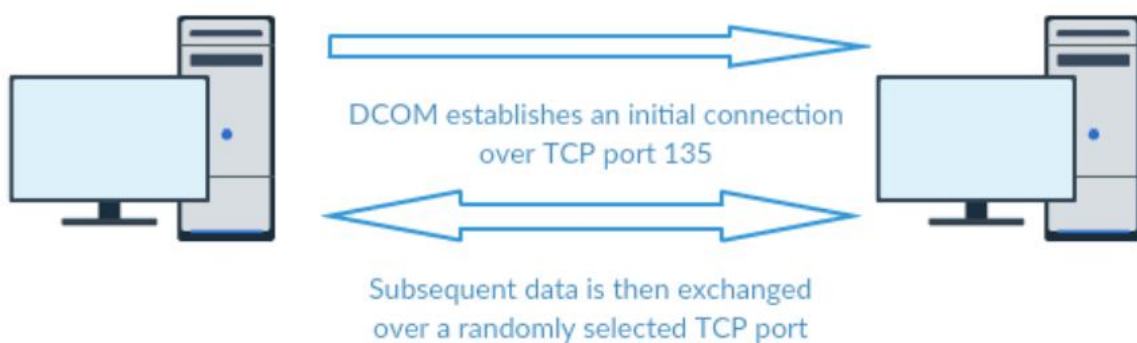


Figure 16: WMI communication between two devices

DCOM establishes an initial connection over TCP port 135.

```

▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.10.11
▲ Transmission Control Protocol, Src Port: 135 (135), Dst Port: 49163 (49163), Seq: 0, Ack: 1, Len: 0
  Source Port: 135
  Destination Port: 49163
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)

```

Subsequent data is then exchanged over a randomly selected TCP port.

```

▶ Internet Protocol Version 4, Src: 192.168.10.11, Dst: 192.168.10.10
▲ Transmission Control Protocol, Src Port: 49167 (49167), Dst Port: 49154 (49154), Seq: 2081, Ack: 1355, Len: 208
  Source Port: 49167
  Destination Port: 49154
  [Stream index: 5]
  [TCP Segment Len: 208]
  Sequence number: 2081 (relative sequence number)
  [Next sequence number: 2289 (relative sequence number)]
  Acknowledgment number: 1355 (relative ack number)

```

Figure 17: WMI communication between two devices via Wireshark

WMI Header with Wireshark



```

▶ Internet Protocol Version 4, Src: 192.168.20.129, Dst: 192.168.10.10
▶ Transmission Control Protocol, Src Port: 50008 (50008), Dst Port: 49155 (49155),
▲ Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fra
  Version: 5
  Version (minor): 0
  Packet type: Request (0)
  ▶ Packet Flags: 0x83
  ▶ Data Representation: 10000000
  Frag Length: 224
  Auth Length: 16
  Call ID: 17
  Alloc hint: 152
  Context ID: 4
  Opnum: 20
  Object UUID: 0001b414-04b4-0000-bbf2-1137bad531fd
  Auth type: NTLMSSP (10)
  Auth level: Packet (4)
  Auth pad len: 8
  Auth Rsvrd: 0
  Auth Context ID: 0
  [Response in frame: 467]
  Stub data: 05000700000000000000000000000000c1c0c83518ccb84aad697067...
  Auth Padding: 0000000000000000
  ▶ NTLMSSP Verifier

```

Figure 18: WMI Header with Wireshark

2.14 Related Works

2.14.1 WMI Classes and Namespaces with Different OS

WMI represents most data related to operating system information and actions in the form of objects. So it important to find number of WMI classes and namespaces with different operating systems.

Here is some information that found with related to Different Microsoft Operating systems
gwmi -namespace root\cimv2 -Recurse -list

Windows PowerShell Get-WMIObject cmdlet has a convenient alias, gwmi

When running gwmi -namespace root\cimv2 -Recurse -list, Windows PowerShell returns all instances in a readable text format

```
PS C:\Users\Administrator> gwmi -namespace root\cimv2 -Recurse -list

NameSpace: ROOT\cimv2

Name                Methods              Properties
----                -
__SystemClass       {}                   {}
__thisNAMESPACE     {}                   {SECURITY_DESCRIPTOR}
__Provider           {}                   {Name}
__Win32Provider      {}                   {ClientLoadableCLSID, CLS...
__ProviderRegistration {}                   {provider}
__EventProviderRegistration {}                   {EventQueryList, provider}
__ObjectProviderRegistration {}                   {InteractionType, provide...
__ClassProviderRegistration {}                   {CacheRefreshInterval, In...
__InstanceProviderRegistration {}                   {InteractionType, provide...
__MethodProviderRegistration {}                   {provider}
__PropertyProviderRegistration {}                   {provider, SupportsGet, S...
__EventConsumerProviderRegistration {}                   {ConsumerClassNames, prov...
__NAMESPACE         {}                   {Name}
__IndicationRelated {}                   {}
__FilterToConsumerBinding {}                   {Consumer, CreatorSID, De...
```

Figure 19: WMI Classes and Namespaces

OS	WMI classes
Windows XP	1000+
Windows 7	1500+
Windows 8	1800+
Windows 10	2000+

Table 03- WMI Classes and Namespaces with operating systems

All the OS included thousands' of WMI classes present. This means that there is a massive volume of retrievable operating system data.

2.14.2 Deep analyses of WMI attacks

A data set is a collection of related, discrete items of related data that may be accessed individually or in combination or managed as a whole entity. when select dataset for analysis that datasets should fit with area of reaserch.so in this case need to create dataset for WMI activity. For identify WMI patterns make a dataset witch include more than 200 WMI queries with TCP dump. normally this TCP Dump consists of more than 60000 packets .so it's not easy analyse those packet with the binary pattern. Then converted all the packet to the hexadecimal format.

For that task use a python script and converted it to the CSV format.

T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
0	0	83	10000000	d000	1000	45010000	84000000	400	1400	07680200t05000700C0a			4	0c	0	0 01t
0	0	83	10000000	d000	1000	68010000	88000000	400	1400	1e940200t05000700C0a			4	8	0	0 01t
0	0	83	10000000	d000	1000	b2010000	88000000	400	1400	23dc0200t05000700C0a			4	8	0	0 01t
0	0	83	10000000	d000	1000	69020000	8c000000	400	1400	205c0200t05000700C0a			4	4	0	0 01t
0	0	83	10000000	e000	1000	e70d0000	a0000000	400	1400	108c0200t05000700C0a			4	0	0	0 01t
0	0	83	10000000	e000	1000	3.00E+01	9c000000	400	1400	0db80200t05000700C0a			4	4	0	0 01t
0	0	83	10000000	f000	1000	1e0e0000	ac000000	400	1400	0af40200b05000700C0a			4	4	0	0 01t
0	0	83	10000000	e000	1000	3.90E+02	9c000000	400	1400	09280200t05000700C0a			4	4	0	0 01t
0	0	83	10000000	d000	1000	5.50E+02	88000000	400	1400	02c40200t05000700C0a			4	8	0	0 01t
0	0	83	10000000	e000	1000	7.20E+02	98000000	400	1400	14480100t05000700C0a			4	8	0	0 01t
0	0	83	10000000	d000	1000	8d0e0000	88000000	400	1400	22840200t05000700C0a			4	8	0	0 01t

Figure 20: WMI pattern analyse with CSV

With the CSV file analyse pattern and find the common signature for the remote WMI packet. When two device communicate over the network using DECRPC protocol there are many WMI packets transmissions happens both end. then its required to identify exact packet that include the WMI execution command.

Here highlighted Colum have same pattern that value equal to the decimal 20 and that is the operation number filed of the DCE RPC header format. When the Operation number field equal to 20 or 24 that Protocol data unit include the WMI command. Also some other same pattern can identify with the csv file, but these are common field for the header structure.as an example version number etc.

Then using above information further analyse the data set with the Wireshark. With Wireshark filter only packets that DCE RPC header operation number field equal to 20 or 24.

When analysing results this time Wireshark only select a 200 packets. But analyse use same dataset that include more than 60000 packets.

The image shows a Wireshark capture window titled 'cap.pcapng'. The filter bar contains the expression 'dcerpc.opnum==20 || dcerpc.opnum==24'. The packet list table below shows four packets:

No.	Time	Source	Destination	Protoc	Length	Info
1703	851.079174	192.168.10.11	192.168.10.10	DCE...	278	Request: call_id: 195
1704	851.079476	192.168.10.10	192.168.10.11	DCE...	310	Response: call_id: 195
2042	1579.8410...	192.168.10.11	192.168.10.10	DCE...	262	Request: call_id: 226
2043	1579.8415...	192.168.10.10	192.168.10.11	DCE...	310	Response: call_id: 226

Figure 21: WMI pattern analyse with wireshark

With this analysis observe that with the operation number of the DEC/RPC header can identify packets that have remote execution code.

Operation number

The operation number is a 16-bit non-negative integer that identifies a particular operation within the interface being called. Op number include inside the byte 63 and 64. and that information important when detecting attack.

- Op Num:20: "ExecQuery",
- Op Num:21: "ExecQueryAsync",
- Op Num:22: "ExecNotificationQuery",
- Op Num:23: "ExecNotificationQueryAsync",
- Op Num:24: "ExecMethod",
- Op Num:25: "ExecMethodAsync"

When opnum (operation number) =20 that PDU request to get information about the remote system

When opnum (operation number) =24 that PDU try to run process on remote host. This command included inside the stub data field in the DCERPC PDU.

- ⊕ Packet Flags: 0x83
- ⊕ Data Representation: 10000000
 - Frag Length: 208
 - Auth Length: 16
 - call ID: 361
 - Alloc hint: 132
 - Context ID: 4

Opnum: 20

0000	08	00	27	fb	63	2b	08	00	27	3f	e1	11	08	00	45	00
0010	00	f8	64	a5	40	00	80	06	ff	f4	c0	a8	0a	0b	c0	a8
0020	0a	0a	c6	0b	c0	02	ed	d3	93	9a	74	e9	d7	5e	50	18
0030	3e	ec	38	9b	00	00	05	00	00	83	10	00	00	00	d0	00
0040	10	00	69	01	00	00	84	00	00	00	04	00	14	00	0b	74
0050	02	00	8c	03	00	00	9a	f3	b7	03	d7	de	4a	b8	05	00

Figure 22: DCE RPC Operation number with Wireshark

Operation Number

After identifying the OPNUM value (20 or 24) then need to feather analyse that packet for identify the attack. Inside the DCE RPC packet there is a field called stub data. That stub data field contain the command that executed by attacker.

IP Header 20 Bytes	TCP Header 20 Bytes	DCE RPC Header 40 Bytes	DCERPC Stub Data
------------------------------	-------------------------------	-----------------------------------	-------------------------

```

Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Reque
  Version: 5
  Version (minor): 0
  Packet type: Request (0)
  Packet Flags: 0x83
  Data Representation: 10000000
  Frag Length: 208
  Auth Length: 16
  Call ID: 361
  Alloc hint: 132
  Context ID: 4
  Opnum: 20
  Object UUID: 0002740b-038c-0000-9af3-b703d7de4ab8
  Auth type: NTLMSSP (10)
  Auth level: Packet (4)
  Auth pad len: 12
  Auth Rsvd: 0
  Auth Context ID: 0
  [Response in frame: 44]
  
```

Stub data (132 bytes)	
0000	08 00 27 fb 63 2b 08 00 27 3f e1 11 08 00 45 00 ..'.c+.. '?....E.
0010	00 f8 64 a5 40 00 80 06 ff f4 c0 a8 0a 0b c0 a8 ..d.@...
0020	0a 0a c6 0b c0 02 ed d3 93 9a 74 e9 d7 5e 50 18t..^P.
0030	3e ec 38 9b 00 00 05 00 00 83 10 00 00 00 d0 00 >.8.....
0040	10 00 69 01 00 00 84 00 00 00 04 00 14 00 0b 74 ..i.....
0050	02 00 8c 03 00 00 9a f3 b7 03 d7 de 4a b8 05 00J..
0060	07 00 00 00 00 00 00 00 00 00 82 59 83 76 b6 31Y.v.1
0070	9a 40 97 5e 14 44 37 f3 bc f5 00 00 00 00 55 73 .@.^.D7.US
0080	65 72 03 00 00 00 06 00 00 00 03 00 00 00 57 00 er.....
0090	51 00 4c 00 00 00 55 73 65 72 19 00 00 00 32 00 Q.L...Us er....2.
00a0	00 00 19 00 00 00 73 00 65 00 6c 00 65 00 63 00s. e.l.e.c.
00b0	74 00 20 00 2a 00 20 00 66 00 72 00 6f 00 6d 00 t.*. .f.r.o.m.
00c0	20 00 57 00 69 00 6e 00 33 00 32 00 5f 00 73 00 .w.i.n. 3.2._.s.
00d0	68 00 61 00 72 00 65 00 00 00 10 00 00 00 00 00 h.a.r.e.
00e0	00 00 00 00 00 00 00 00 00 00 00 00 00 0a 04
00f0	0c 00 00 00 00 00 01 00 00 00 7b 22 46 79 5b 29{"Fy[]
0100	5b 51 07 00 00 00 [Q....

Figure 23: DCE RPC stub data with Wireshark

Command that execute by attacker

CHAPTER 3 – DESIGN

3.1 Security Implementations of Networks

Network administrators cannot be successfully managed all the security challenges by any single application. While implementing device hardening, authentication, authorization, and accounting (AAA) access control, and firewall features are all part of a properly secured network, these features still cannot secure the network against fast-moving Internet worms and viruses.

Intrusion prevention is required throughout the entire network to successfully detect and stop an attack at every inbound and outbound point.

Once implementing IDS or IPS, it is important to be familiar with the types of systems available and the placement of these systems.

With this research outcome pattern based intrusion detection system appropriate for detect vulnerable WMI incident inside the network. When designing IDS software solution, can use common identified WMI patterns in the research. This is done with support of different software applications. Also Prepare a detailed log report on the captured events is essentials. Then System Administrators can use these log reports to analyze vulnerable activity inside the network.

3.1.1 Promiscuous Monitoring in Network

Port mirroring is a feature that allows a switch to make a duplicate copy of an incoming Ethernet frame, and then send it out a port with a packet analyser attached for capture. The original frame is forwarded in the usual manner.

A managed switch that supports port mirroring, this feature allows you to configure the switch to redirect the traffic that occurs on some or all ports to a specific designated monitoring port on the switch. Then this port ideal device for network monitoring purposes. The way port mirroring is configured depends on the specific vender and configuration can differ with different vender devices

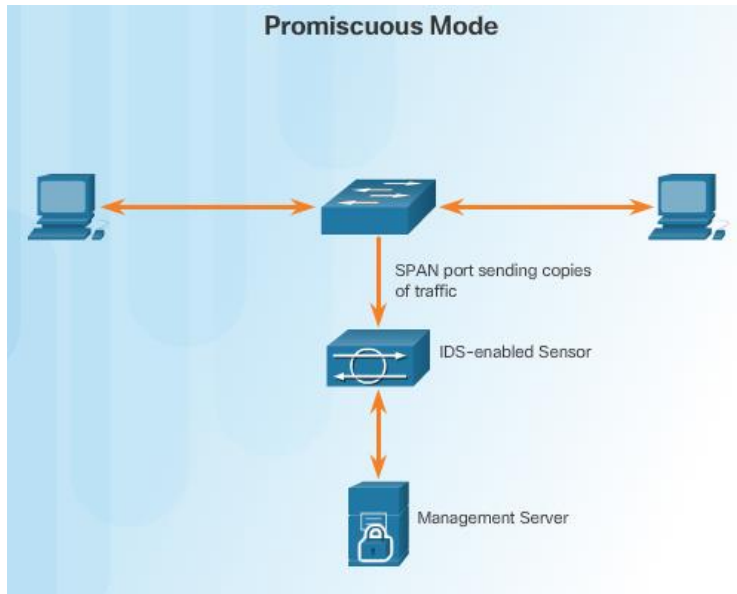


Figure 24: Promiscuous Monitoring in Network

3.1.2 WMI Attacks Categorization

In this IDS /IPS designing stage defined attacks in to two category or two severity levels. That is Critical and Medium.

Severity Level: Critical

WMI Vulnerabilities that under the critical range usually have most of the following characteristics:

- Try to start or end a process on remote host
- Try to drop file to the remote host
- Try to change registry settings on remote host
- Try to remove or modify files on remote host etc.

Severity Level: medium

WMI Vulnerabilities that under the medium range usually have most of the following characteristics:

- Try to get remote system information
- Ex: operating system; user accounts

3.2 Design WMI Intrusion Detection System(IDS)

IDS Design Flowchart

The network must be able to identify incoming malicious traffic in order to stop it. A signature is a patterns that an IDS and an IPS use to detect typical intrusion activity. For this design used atomic signature that is simplest type of signature.

As sensors scan network packets, they use signatures to detect known attacks and respond with predefined actions. A WMI malicious packet flow has a specific type of activity and signature. An IDS sensor examines the data flow using many different signatures. A sensor takes action when it matches a signature with a data flow, such as logging the event or sending an alarm to the IDS or IPS software.

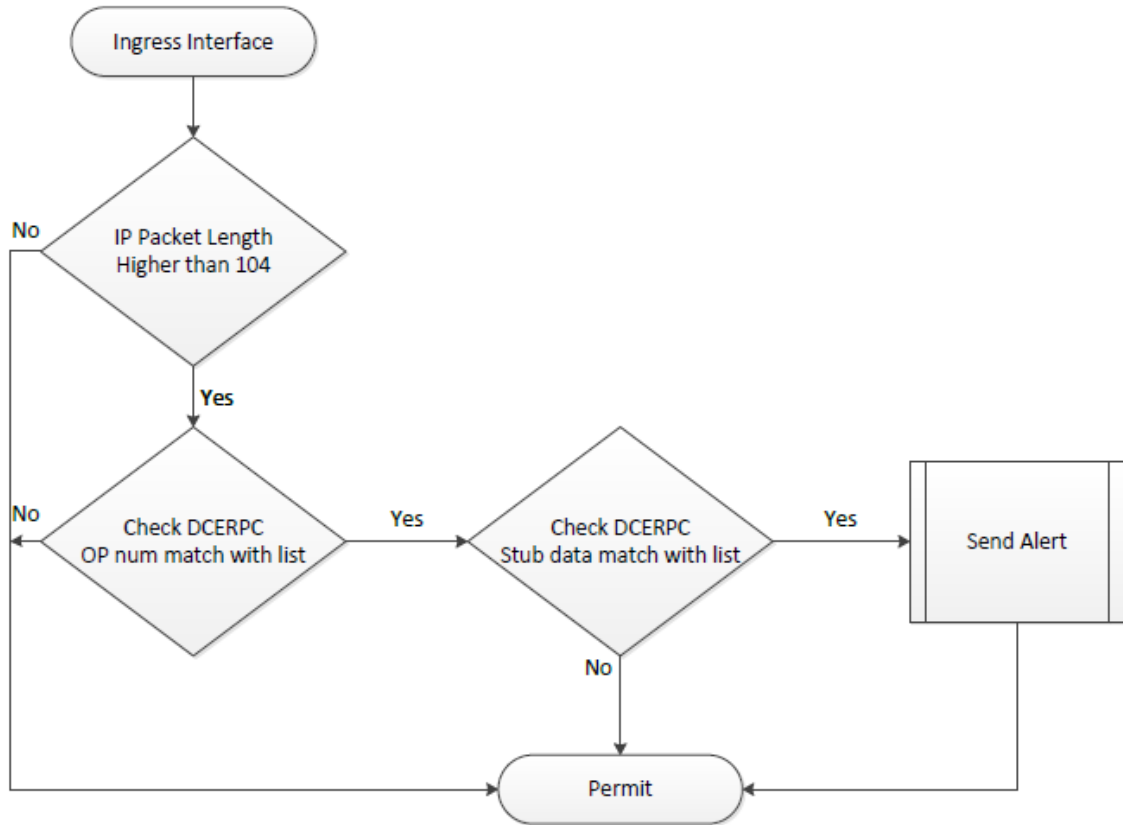


Figure 25: IDS Design Flowchart

Step1:

Before process the packet fist need to verify that packet has minimum length. If the minimum length not fulfil that packet no need to observe feather more.as a minimum length, According to the headers and trailer length minimum length should be more than 104 bytes. If the length less than 104 that cannot be include WMI command inside the packet and that means that cannot be a WMI execution. This is the way that 104 value got.

$$ipheader[20] + tcpheader[20] + dcerpcheader[40] + stubdata[0] + auth[8] + NTLMSSPVerifier[16]$$

This is easily can detect with the IP header total Length field.

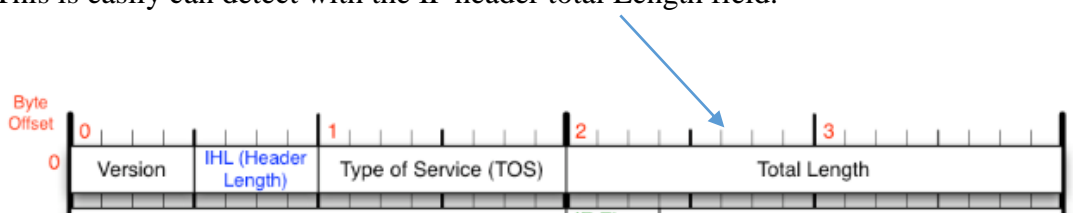


Figure 26: IP header total Length field

Step2:

With the research identify that when Operation number equal to value 20 and 24 there are some WMI activity happens. So this will check the 2-Byte length area inside the DCERPC header in-between 44 and 46 bytes. With the hexadecimal decimal 20 represent as 14 and decimal 24 represent as 18.

Step3:

As sensors scan network packets, they use signatures to detect known attacks and respond with predefined actions. A malicious packet flow has a specific type of activity and signature. when ingress traffic pattern equal to the signature database that means attack signature is detected. Then need to Generate an alert and Log the activity

3.3 Design WMI Intrusion Protection System(IPS)

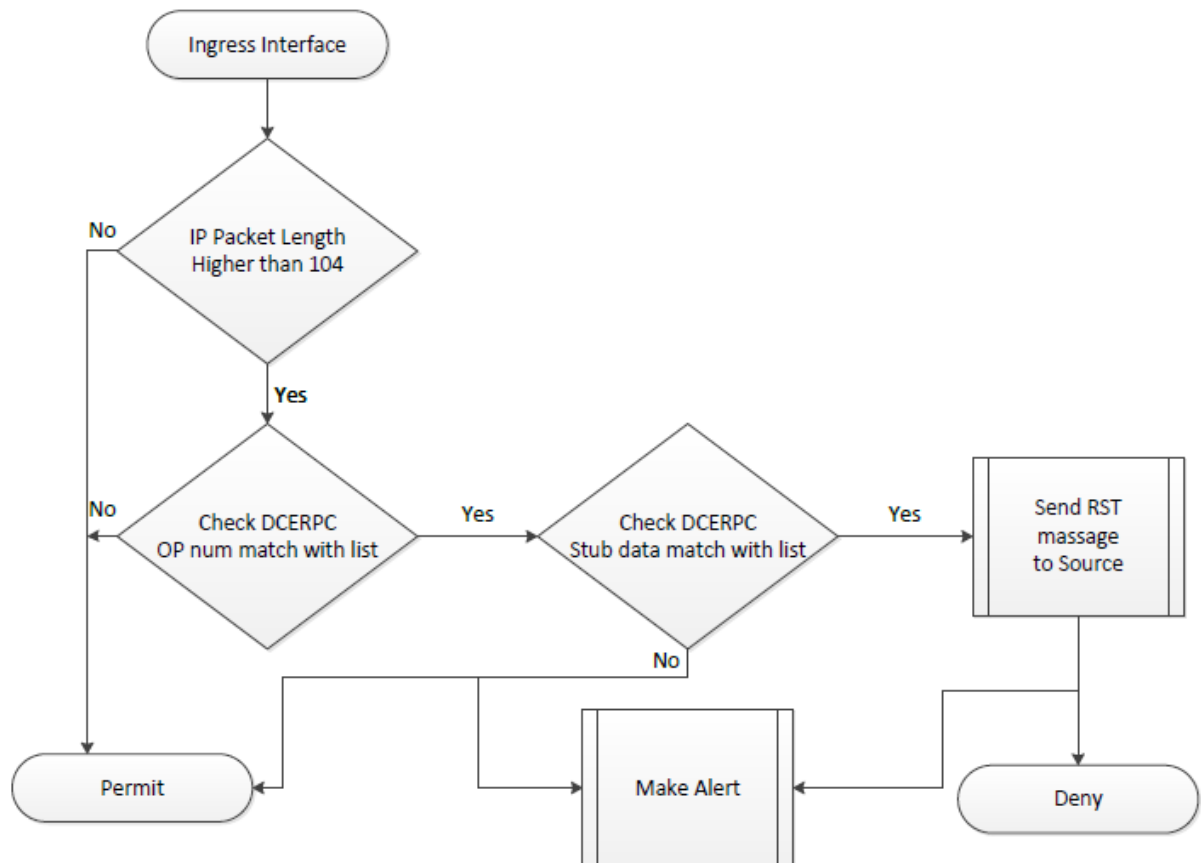


Figure 27: WMI Intrusion Protection System flow chart

Step 1,2 and 3 same as the IDS process described in IDS Section.

IPS devices are designed to stop or prevent unwanted traffic on a network. This section describes main Functionality of the WMI based IPS. When a signature detects the activity for which it is configured, the signature triggers one or more actions. Several categories of actions can be invoked,

- Generate an alert.
- Log the activity.
- Drop or prevent the activity.

- Reset a TCP connection.
- Allow the activity.

The available actions depend on the signature type. There are 3 main actions taken by of the implemented WMI based IPS

Deny Traffic
Reset Traffic
Allow Traffic

Blocking Activity

WMI Network-based IPS devices usually provide this blocking functionality when an attack signature is detected. also send a log message and make alert.

Resetting a TCP Connection

The TCP Reset Signature Action is a basic action that can be used to terminate TCP connections by generating a packet for the connection with the TCP RST flag set. IPS devices use the TCP reset action to abruptly end a TCP connection that is performing unwanted operations. The reset TCP connection action can be used in combination with deny packet and deny connection actions. Deny packet and deny flow actions do not automatically because TCP reset actions to occur.

Allowing the Activity

The final action is the Allow Signature action. WMI Network-based IPS devices provide this allowing functionality when ingress traffic pattern not equal to the signature database or prevent list. also send a log message and make alert.

CHAPTER 4 - IMPLEMENTATION

4.1 WMI Intrusion Detection System(IDS) with snort

These are the rules that used to detect WMI activity inside the network.

Snort rules;

```
alert tcp any any -> any [135:] (msg:"ExecQuery"; dce_opnum:20; dce_stub_data;
pcre:"/^\.{12}(\x00\x00\x00\x00|\.{12})/s"; sid:1000068;)
```

```
alert tcp any any -> any [135:] (msg:"ExecMethod"; dce_opnum:24; dce_stub_data;
pcre:"/^\.{12}(\x00\x00\x00\x00|\.{12})/s"; sid:1000072;)
```

this rule checks all TCP packets form any source to any IP destination with 135 destination port number and also check that containing OP number field equal to 20 or 24 and then make alert.

pcre: Perl compatible regular expressions

execute some command with the attacker pc and try to detect this WMI activity with the snort.

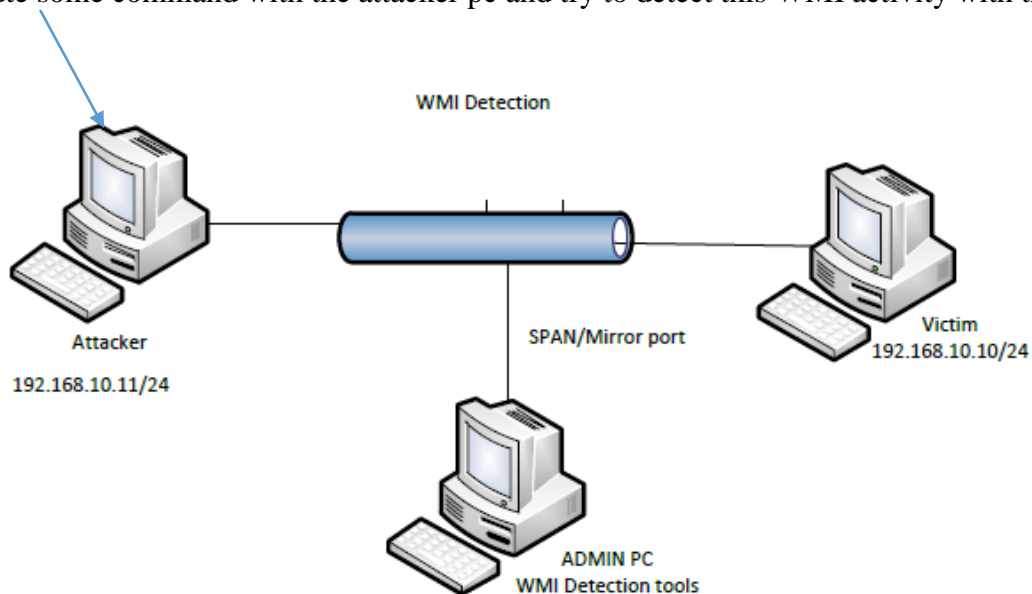


Figure 28: WMI Intrusion Detection System(IDS) with snort

```
PS C:\Users\Administrator> Get-WmiObject Win32_share -ComputerName 192.168.10.10 -Credential %computername%
r
Name Path Description
----
ADMIN$ C:\Windows Remote Admin
C$ C:\ Default share
IPC$ Remote IPC

PS C:\Users\Administrator> Invoke-WmiMethod -Class Win32_Process -Name Create -ArgumentList 'notepad.exe'-C
192.168.10.10 -Credential %computername%\administrator

GENUS : 2
CLASS : __PARAMETERS
SUPERCLASS : __PARAMETERS
DYNASTY : __PARAMETERS
RELPATH :
PROPERTY_COUNT : 2
PERMISSION :
```

This execution cannot detect with snort

Figure 29: WMI Intrusion Detection System(IDS) snort output

```

milan@wmi:~/Desktop$ sudo snort -q -A console -d -l /var/log/snort/ -c /etc/snort/snort.conf -u snort -g snort -i ens33
02/16-12:37:03.479516  [**] [1:1000068:0] ExecQuery [**] [Priority: 0] {TCP} 192.168.10.11:49222 -> 192.168.10.10:49155

```

sudo snort -q -A console -d -l /var/log/snort -c /etc/snort/snort.conf -snort -g snort -i ens33

sudo - (Snort requires root permissions to change the interface mode into promiscuous. Due to that here command starts with sudo)

-q Tells Snort to run quietly. Does not display banner and initialization information. If you aren't interested in the initialization messages, you can suppress them with this

-A alert-mode

Generates an alert using one of the specified alert-modes: fast, full, none. Rather than specifying the alert mode within a configuration file, we can include it here at the command line.

-d Displays the application layer data when in verbose or packet logging mode

-l logging-directory

Specifies the logging directory. All alerts and packet logs are placed in this directory.

-c config-file

Allows you to specify which configuration file you want to use. If we have different configurations with various rules enabled, we can specify which configuration to use at the command line. This option is required when Snort is run in NIDS mode.

-i interface

Specifies which interface Snort should listen on. This option is used on machines that have more than one network interface card or that have different kinds of interfaces, besides Ethernet. Naming conventions for interfaces vary between operating systems.

-u user

Changes the default user ID or UID under which Snort runs after initialization.

With the Snort not potential to detect all vulnerable WMI activities. Snort will only detect WMI remote query and not detect remote execution. According to the testing result, we can say that snort can only detect WMI incident related to the OP Number 20 and cannot detect remote executions that related to the Op Number 24.

4.2 Implementation of WMI based IDS using python

These are the main functionalities of implemented python based IDS.

Step 1:

if pkt['IP'].len > 104:

this will check that frame minimum length. According to the headers and trailer length minimum length should be more than 104 bytes. If the length less than 104 that cannot be a DCERPC packet and that means that cannot be a WMI execution. This is the way that 104 value got.

Step2:

data = pkt['Raw'].load.encode('hex')

to analyses the DCERPC header now need to decode the packet header in to hexadecimal format.

This is essentials because need to identify this packet include WMI command or no not.

if len(pkt['Raw']) > 64 and data[0:4] == '0500'

if data[44:46] == '14' or data[44:46] == '18':

this is check that DCERPC version equal to the 5 and Operation number value with hexadecimal.

With the research identify that when Operation number equal to value 20 and 24 there are some WMI activity happens. So this will check the 2-Byte length area inside the DCERPC header in-between 44 and 46 bytes. With the hexadecimal decimal 20 represent as 14 and decimal 24 represent as 18.

Step3

d = pkt['Raw'].load

This function based on scapy module to handle packet generate and analyse. Also based on tcpdump tool to sniff packets. In this module, define some basic type packets which can simple used by name.

regex = re.compile('[^a-zA-Z]')

defining regular expression. This would match any English alpha characters.

msg = re.sub(regex, '', d[68:-24])

This will remove unwanted symbols and rearrange data

This searches for a specific and pre-defined pattern. A signature-based IDS sensor compares the network traffic to a database of known attacks, and triggers an alarm if a match is found. The signature trigger is a textual output.

00050 Thu Sep 8 07:40:32 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f
192.168.10.10 Win32_LoggedOnUser

00051 Thu Sep 8 07:40:47 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f
192.168.10.10 Win32_Desktop

Log file record format

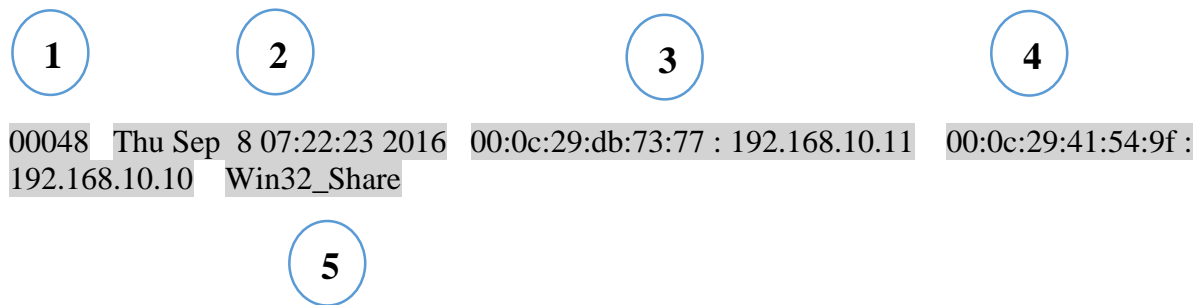


Figure 32: log file structure

1	Sequence no	Log message unique sequence number
2	Timestamp	Time that command executed
3	Source	IP Address and MAC Address of Source device
4	Destination	IP Address and MAC Address of Destination device
5	Command	Command that executed by source device

4.3 Implementation of WMI based IPS using python

These are the main functionalities of implemented python based IPS.

Before process the packet first need to verify that packet has minimum length. If the minimum length not fulfil that packet no need to observe further more as a minimum length, According to the headers and trailer length minimum length should be more than 104 bytes. If the length less than 104 that cannot be include WMI command inside the packet and that means that cannot be a WMI execution. This is the way that 104 value got.

```
ipheader[20] + tcpheader[20] + dcerpcheader[40] + stubdata[0] + auth[8] + NTLMSSPVerifier[16]
```

```
if ip_length > 104:
```

```
    tcp = packet[20:]
    t_s_port, t_d_port, t_seq_num, t_ack_num, t_info, t_flags, t_win,
t_checksum, t_u_ptr = \
        struct.unpack(">HLLsBHHH", tcp[0:20])
    d_version, d_p_type, d_p_flags, d_d_rep, d_f_len, d_a_len, d_c_id,
d_a_hint, d_con_id, d_op_num = \
```

```
struct.unpack("<HBB4sHH4s4sHH", tcp[20:44])
```

if ip_length > 104:

The struct module is designed to unpack heterogeneous data to a tuple based on a format string. It makes more sense to unpack the whole struct at once rather than trying to pull out one field at a time

```
struct.unpack(">HLLsBHHH", tcp[0:20])
```

this will unpack the TCP header to the normal format with the field

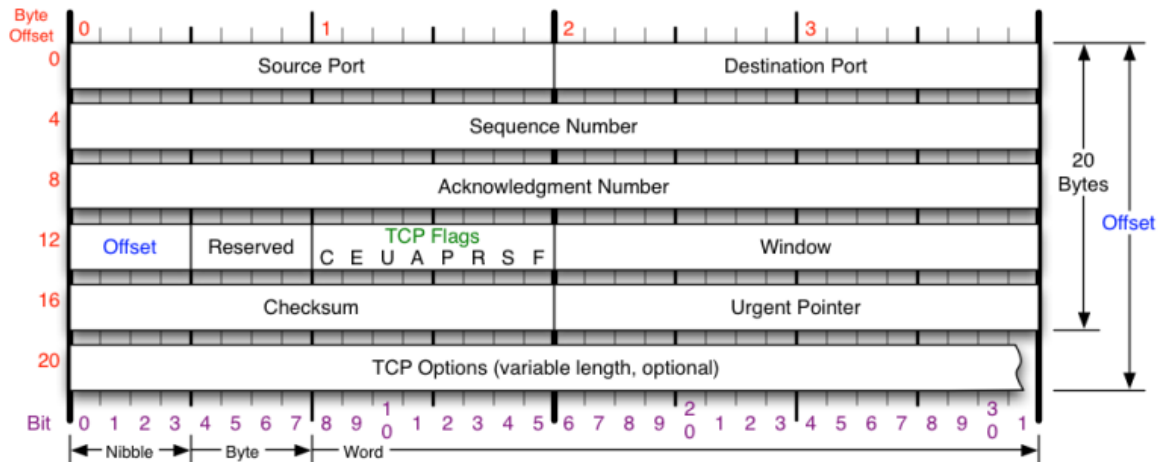


Figure 33: TCP header

```
d_version, d_p_type, d_p_flags, d_d_rep, d_f_len, d_a_len, d_c_id, d_a_hint, d_con_id,
d_op_num = \
struct.unpack("<HBB4sHH4s4sHH", tcp[20:44])
```

Now WMI firewall unpack the DCE RPC header using the same mechanism as previously done. That why this unpack byte range from 20 to 44. because DCE RPC header contain 24 Bytes. This is very important because after this firewall need to analyse DCERPC header for identifying WMI activity.

```
if d_version == 5 and d_op_num in op_list.keys():
    regex = re.compile("[^a-zA-Z1-9_]")
    strings = re.sub(regex, '', packet[122:-24]).lower().split()
    if any(map(lambda x: x in strings, prevention_list)):
        match = prevention_list[map(lambda match: match in strings,
prevention_list).index(True)]
```

then firewall check that version value of DCERPC equal to 5 and also check that DCERPC OP-number, if the op-number is equal to list value that define in the firewall then firewall decide that packet include WMI activity.

Then firewall lookup that stub data field in that packet with the prevention list signatures. If firewall find same pattern inside the packet, then that packet drop by the firewall. But there is a problem happens when drop the packets. because this is TCP communication from attacker PC it will retransmission.

In this case firewall regenerate TCP packet and set control flag to RST and that send to the source that generate WMI query

Finally, a connection may be abruptly closed by firewall sending a RST segment to the other. Normally RST segments are sometimes used by firewalls to end suspicious TCP connections or by hosts to actually refuse connection demands. In this scenario this firewall also uses same mechanism.

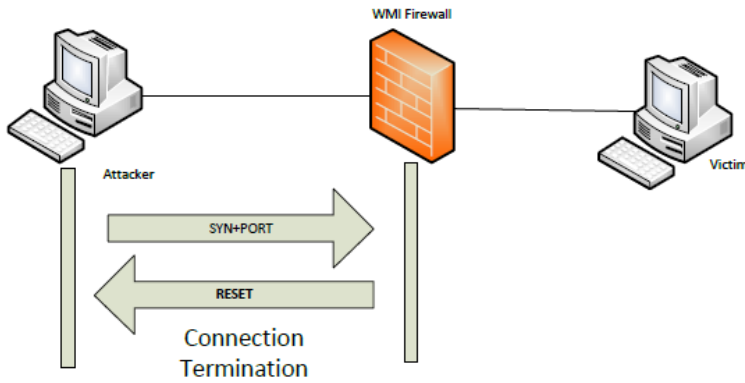


Figure 34: connection reset with WMI IDS

WMI IPS Interface

This is the implemented WMI IPS graphical user interface. With this interface user can specify what are the network interfaces that will need to analyse with the IPS tool

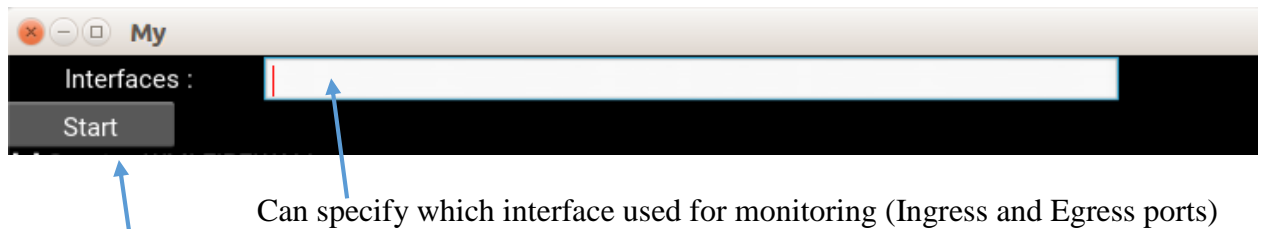
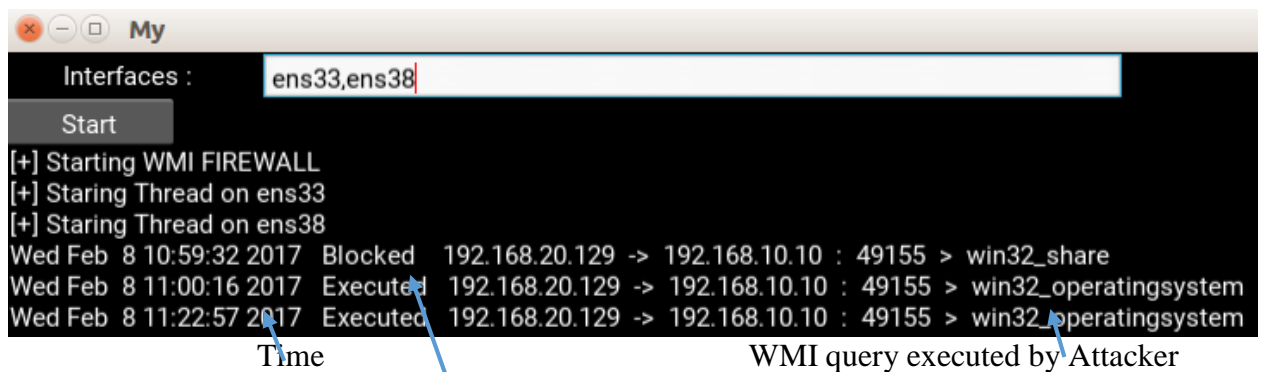


Figure 35: WMI Firewall Interface



This WMI query was drop by firewall and end TCP session with RST flag

Figure 36: WMI Firewall Interface with details

CHAPTER 5 - RESULTS ANALYSIS AND DISCUSSION

Mitigations Technics form WMI Attacks

there are several mitigations technics for prevent some or all WMI attacks from occurring.

1. Disabling the WMI service. Consider your organizations need for remote WMI access. Do consider however any unintended side effects of stopping the WMI service.
2. Blocking the WMI protocol ports. If there is no legitimate need to use remote WMI.
3. WMI, DCOM, and WinRM events are logged to the following event logs:
 - a. Microsoft-Windows-WinRM/Operational
 - b. Microsoft-Windows-WMI-Activity/Operational
 - c. Microsoft-Windows-DistributedCOM

Disabling WMI is mitigation technologies but then network administrators cannot use WMI for their administrative tasks but blocking DCOM port cannot done all the times because port 135 used by lots of other services.

Result Analysis and Future Works

With the implemented IPS and IDS tool detect WMI related activity and report them to administrator and or drop the packets. But using this tool we cannot secure entire network form all types of attacks. because this can detect WMI based attacks only. for this implementations used atomic signatures that consumes minimal resources, such as memory, on the IPS or IDS device. These signatures are easy to identify and understand because they are compared to a specific event or packet. Traffic analysis for these atomic signatures can usually be performed very quickly and efficiently. These atomic signatures do not maintain status.

A primary advantage of an WMI IDS platform is that it is deployed in offline mode. Since the IDS sensor is not inline, it has no impact on network performance. It does not introduce latency, jitter, or other traffic flow issues. In addition, if a IDS sensor fails it does not affect network functionality. It only affects the ability of the IDS to analyse the data.

However, there are many disadvantages of deploying an IDS platform. An WMI IDS sensor is primarily focused on identifying possible incidents, logging information about the incidents, and reporting the incidents. These incident report based on the signature database of the IDS. according to the research already find more than thousands WMI namespace available in all Microsoft operating systems.so it's very hard to include all the pattern related to namespace.

An IPS be configured to perform a packet drop to stop the attack, and stop that session with sending RST packets to the source. Additionally, because IPS sensors are inline, then this effect to the network performance. A disadvantage of IPS is that errors, failure, and overwhelming the IPS sensor with too much traffic can have a negative effect on network performance. An IPS can affect network performance by introducing latency and jitter.

This IPS observe all the packets that have packet length 104 or more.so that time-sensitive applications, such as VoIP is affected to analyse and then can have some delay. This implementation uses atomic signatures that is not maintain state. A composite signature

(stateful signature) identifies a sequence of operations distributed across multiple hosts over an arbitrary period of time. Unlike atomic signatures, the stateful properties of composite signatures usually require several pieces of data to match an attack signature, and an IPS device must maintain state. This can be better solution for implemented firewall and wish to improve the IDS and IPS with composite signature in the future.

Implemented WMI IPS system log file system format same as snort log file system. also in some cases each log possibly requiring separate monitoring or other analysis. Furthermore, i have analyses this WMI IPS generated log file with the standard industry used log analyzer(Splunk). Splunk is well-known log analyzing application used by industry.

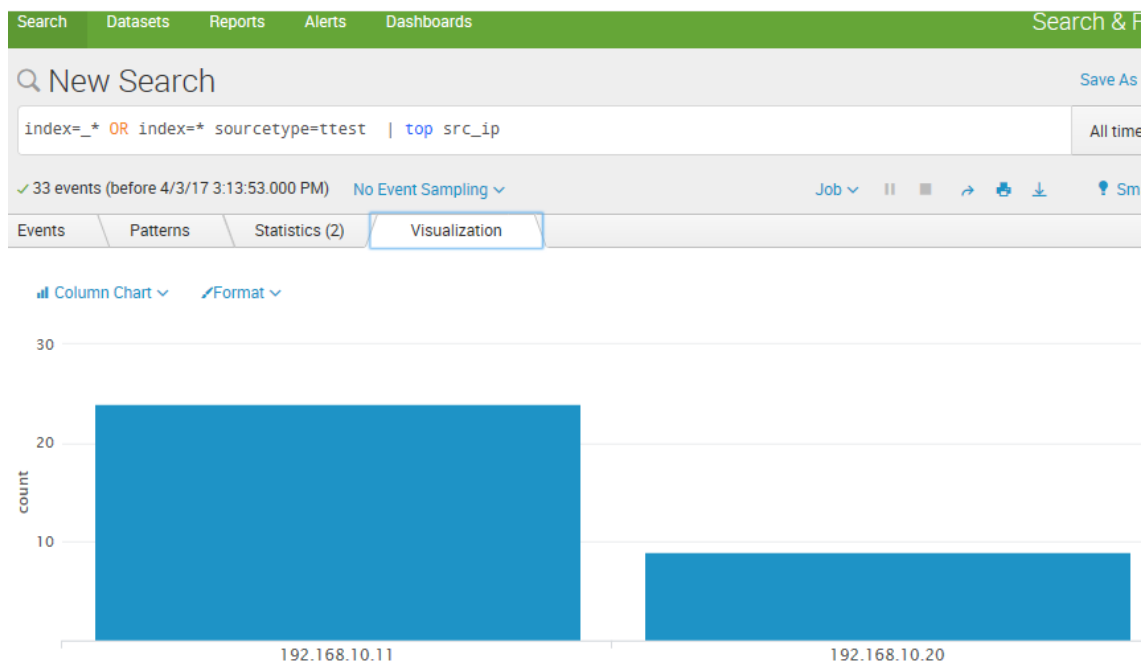
Below figure show the log file that used to analyze with splunk

```
00051 Thu Sep 8 07:40:47 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f 192.168.10.10 Win32_Desktop
00052 Thu Sep 8 07:47:45 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f 192.168.10.10 Win32_TimeZon
00053 Thu Sep 8 08:21:32 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f 192.168.10.10 Win32_Logged0
00053 Thu Sep 8 08:22:23 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_QuotaSe
00054 Thu Sep 8 08:22:40 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Binary
00055 Thu Sep 8 08:23:32 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Securit
00056 Thu Sep 8 08:23:43 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Process
00057 Wed Sep 21 04:53:40 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Share
00058 Wed Sep 21 04:53:42 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Logged0
00059 Wed Sep 21 04:55:23 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Operati
00060 Wed Sep 21 04:57:20 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f 192.168.10.10 Win32_Share
00061 Wed Sep 21 05:00:32 2016 00-50-56-C0-00-01 192.168.10.20 00:0c:29:41:54:9f 192.168.10.10 Win32_Startup
00062 Wed Sep 21 05:01:13 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_BootCon
00063 Wed Sep 21 05:02:50 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Share
00064 Wed Sep 21 05:03:50 2016 00:0c:29:db:73:77 192.168.10.11 00:0c:29:41:54:9f 192.168.10.10 Win32_Share
```

Figure 37: log file generated by WMI IDS tool

in this case first analyze what are the source ip address of that execute WMI remote queries.

Below figure show the results;



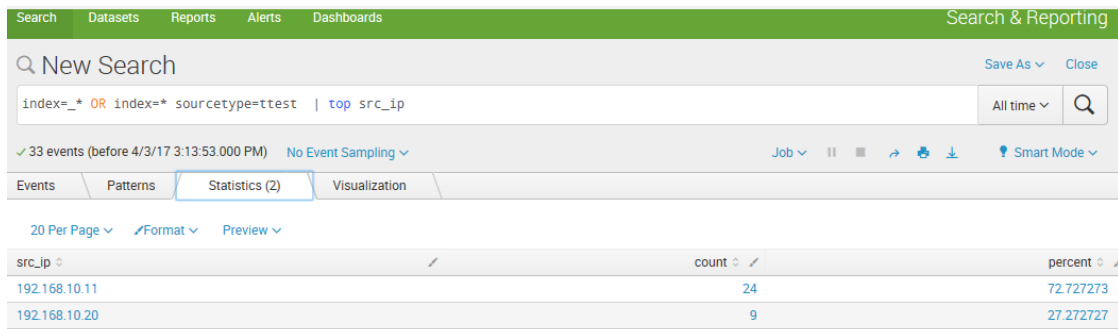


Figure 38: log file analysis with splunk

in this section analyse the specific query related information with log analyse tool.

Below figure show that who execute the win32_share WMI query according to the log file data.

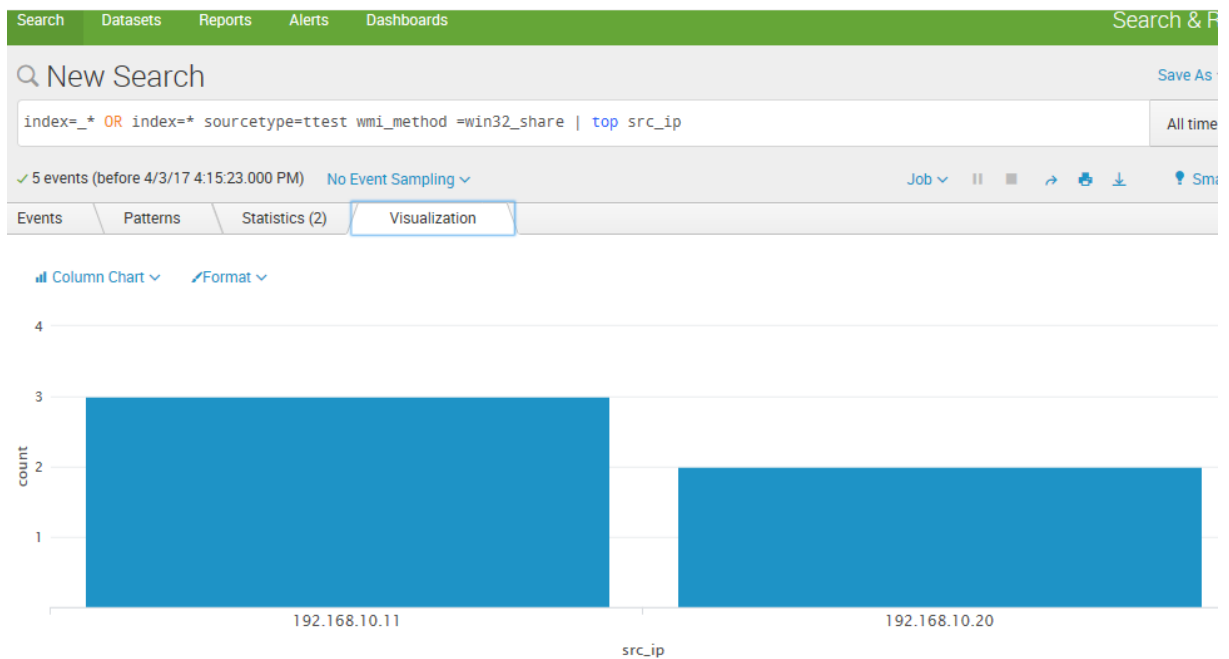


Figure 39: log file analysis with splunk

With the above analyses done with splunk using generated log file can say that log file implementation was comply with standard log reporting format.

A false positive alarm is an expected but undesired result. A false positive occurs when an intrusion system generates an alarm after processing normal user traffic that should not have triggered an alarm. With the implemented IPS/IDS false positive alarm can happens in few situations. Because if administrator try implement Group policy with the WMI filtering this is normal traffic. But this intrusion system can detect this.

A false negative is when an intrusion system fails to generate an alarm after processing attack traffic that the intrusion system is configured to detect. This can happen in special situation. If attacker can change the MTU value of the system before attack, then attacker execution code goes as a small fraction. Then implemented intrusion system cannot detect this with the signatures.

A true positive alarm describes a situation in which an intrusion system generates an alarm in response to known attack traffic. With this implemented tool checked with dataset that included 60000 packets and that dataset include some malicious WMI activity. This implemented intrusion system detects all the malicious WMI activity. so this have very high true positive rate.

A true negative describes a situation in which normal network traffic does not generate an alarm. As the observation and testing done can say that this have very high true negative rate.

CONCLUSION

Windows Management Instrumentation (WMI) In the Windows operating system is powerful technology to manage local and remote systems. WMI is a default service installed on Windows XP and Server 2003 and all Microsoft Operating System. There are many ways to interact with WMI. PowerShell can use easily when interacting with WMI remotely

With WMI easily can perform system reconnaissance, code execution, lateral movement, anti-virus and virtual machine detection, persistence, and data theft.

WMI use Distributed Component Object Model(DCOM) as its default protocol for communication over the network. Distributed Component Object Model(DCOM) establishes an initial connection over TCP port 135. Subsequent data is then exchanged over a randomly selected TCP port using Distributed Computing Environment/Remote Procedure Call (DCE/RPC) protocol. There are several mitigations that may prevent WMI attacks from occurring. These are close the port 135 or disable WMI in the hosts. But problem was then network administrators also cannot use this service for their tasks. Also port 135 not only used for the WMI communication. This port used by many other Microsoft services.so it's impossible to close this port with the systems.

Also I have done some research about the number of available WMI classes on each Microsoft Operating systems. WMI represents most data related to operating system information and actions in the form of objects. So it important to find number of WMI classes and namespaces with different operating systems. All the OS included thousands' of WMI classes present. (windows XP 1000+, windows 7 1500+ windows 8 1800+, windows 10 2000+) This means that there is a massive volume of retrievable operating system data available with all OS.

IDS tool helps the Security /network Administrator to identify the remote access WMI activities that are vulnerable or are attacked more frequently. Also with the IPS tool can stop attack and gain better security.

So this required IDS/IPS based solution for protect network from this type of WMI attacks. before design IPS or IDS it's important to analyse Remote WMI process well. In this research first create dataset related to the WMI activity. This dataset includes more than 20,000 packets and inside that already include 200 packets that related to WMI remote execution activity.

Then analyse the packets and find the common pattern for identify WMI execution process. It's difficult to analyse data set with binary format.so convert that to hexadecimal format using python script. then that convert to the CSV format and observed the patterns. With that observation find the way to detect the specific packets only that include command executed by the user for remote host.

Remote WMI communication use Distributed Computing Environment/Remote Procedure Call (DCE/RPC) as middleware system for execute between existing operating system and remote operating system. Main objective of the DCERPC is to make it possible for a client to access a remote service by simply calling a logical procedure. When someone execute WMI command for remote host that communication can uniquely identify with the DCERPC header value called Operation number. When Operation number value become 20 to 24 range that means there is remote execution happens. When someone try to query information form remote host, Operation number equal to 20 and also someone try to make remote execution (start process) the Operation number become 24.

Then designing structure for the develop IPS and IDS tool considered above information. When we consider the intrusion detection software design this first check the total length of data packet. Because if process all the data packet that will badly effect to the network performance. we consider the packet length should be more than 104 bytes. Because when remote execution happens command that executed by the attacker goes with the stub-data field in the DCE/RPC header. So with the minimum length value can decide that can include WMI execution command or not.

This minimum length value 104 get this way (ipheader[20] + tcpheader[20] + dcerpcheader[24] + uuid[16] + stubdata[0] + auth[8] + NTLMSSPVerifier[16] =total 104)

Then Intrusion detection/protection software check that packet is related to WMI activity or not. This can have done with analysing location 44-46-byte information of the data packet. If that is in between 20 -25 that can include WMI activity.so need to further observation for that packet. This can be malicious WMI query. Than this will compare the stud data field value with the signatures of the IPS/IDS software. If it is match this will generate notification with the severity level and same time make log file with the all required information and make necessary action.

IPS devices are designed to stop or prevent unwanted traffic on a network. When a signature detects the activity for which it is configured, the IPS make one or more actions. Including Generate an alert., Log the activity, Drop or prevent the activity, reset a TCP connection., Allow the activity.

With the WMI based IPS if packet signature match with the signature in the IPS it will drop the packet and stop the communication. But this WMI process is based on TCP most of the time. So after IDS drop the WMI packets again retransmission happen. because of the reliability factor of the Transmission control protocol. Anyway need to stop this.so then IDS software tool reconstruct packet with reset flag and send it to the source. Then that communication forcedly stops with reset packet and no more retransmissions. Also this IPS software pass the notification about the WMI activity and also generate log file for future reference. For this implementation used an atomic signature type. It consists of a single packet, activity, or event that is examined to determine if it matches a configured signature. If it does, an alarm is triggered, and a signature action is perform.so this analyses each and every packets that passes the above stages. This type of signature not care about the communication status.

Log file format of the IPS/IDS software is very important. because it need to be comply with the standard format. Otherwise this log file cannot analyse with the common log analysers. This log format has most in common with the standard log format, which contains data such as the Internet Protocol (IP) address, Mac Address, user query, and access time, among other fields. When designing IPS/IDS software used a logging standard defined by Internet Engineering Task Force (IETF) in RFC 5424 and also that comply with snort IDS log format. Most of the time administrator need to analyse this data in order to obtain beneficial information related to security. Also WMI IPS /IDS generated log file analyse with well-known log analysing applications and so can say that implementation was comply with standard log reporting format.

This implemented Intrusion software tested with the dataset that earlier used and got very high true positive rate. Finally, implemented IPS/IDS tool can help to network /Security administrators to protect their network with WMI based attacks.

References

Online Sources

- [1] <http://www.securityweek.com/attackers-increase-use-powershell-wmi-evade-detection-mandiant>
- [2] [https://msdn.microsoft.com/en-us/library/aa394582\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx)
- [3] [https://msdn.microsoft.com/en-us/library/aa384642\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa384642(v=vs.85).aspx)
- [4] <http://www.dcerpc.org/>
- [5] <https://technet.microsoft.com/en-us/library/cc958799.aspx>
- [6] <https://technet.microsoft.com/itpro/powershell/windows/cim/index>
- [7] <https://www.snort.org/>
- [8] <https://www.python.org/>
- [9] <https://kivy.org/#home>
- [10] <http://www.secdev.org/projects/scapy/>
- [11] <https://msdn.microsoft.com/en-us/powershell/mt173057.aspx>

Handbooks

- [12] Justin Seitz, Black Hat Python: Python Programming for Hackers and Pentesters, chapter4
- [13] Data & Computer Communications, William Stallings

Reports

- [14] Ballenthin-Graeber-Teodorescu, WMI-Attacks-Defence-Forensics
<https://media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/DEFCON-23-Ballenthin-Graeber-Teodorescu-WMI-Attacks-Defense-Forensics.pdf>
- [15] Christopher Glycer, Devon Kerr ,There's Something About WMI ,October 7, 2014 ,
https://files.sans.org/summit/Digital_Forensics_and_Incident_Response_Summit_2015/PDFs/TheresSomethingAboutWMIDevonKerr.pdf
- [16] Matt Graeber ,Abusing Windows Management Instrumentation (WMI) to Build a Persistent, Asynchronous, and Fileless Backdoor,
<https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor.pdf>