Cyber Security Portfolio Ass2

COIT11241: Cyber Security Technologies

Name: Aidan Petre

ID: 12141971

Campus: Cairns

Contents

[BumbleBee Summary 2](#_Toc135593620)

[T1047 – Windows Management Instrumentation 2](#_Toc135593621)

[T1566 – Phishing 2](#_Toc135593622)

[T1548.002 – Abuse Elevation Control Mechanism: Bypass User Account Control 2](#_Toc135593623)

[T1041 – Exfiltration Over C2 Channel 2](#_Toc135593624)

[T1053.005 – Scheduled Task/Job: Scheduled Task 2](#_Toc135593625)

[Attack Emulation 3](#_Toc135593626)

[Scripts 3](#_Toc135593627)

[T1047 3](#_Toc135593628)

[T1053.005 3](#_Toc135593629)

[T1548.002 3](#_Toc135593630)

[Detections/Safeguards 3](#_Toc135593631)

[T1047 3](#_Toc135593632)

[T1053.005 3](#_Toc135593633)

[T1548.002 3](#_Toc135593634)

# BumbleBee Summary

There is a large number of sub/techniques that can be used by the BumbleBee software, a few of those techniques are more important than others. The simplest way to find these key techniques is to ask, “If this technique is no longer used can the software still be successful”. The following techniques are all required for the software.

## T1047 – Windows Management Instrumentation

All malicious software requires a form of execution. WMI is a feature that gives environment access both locally and remotely. This means it can give attackers a way to execute the malicious files on target machines remotely. Without this technique BumbleBee could sit idle on a machine and cause no harm as nothing will be executed.

## T1566 – Phishing

BumbleBee uses two of the sub techniques withing T1566 however I will group them in one as they serve the same purpose of phishing just different scenarios. T1566.001, spearphishing attachment and T1566.002 which is in simple terms having a malicious file attached to the email, spearphishing link which is similar to the prior mention sub technique except a link is used instead of an attachment, are the two sub techniques that are used by the software. I have selected these sub techniques as examples for how the software can arrive on the target machine, there are other sub techniques that may also be used. BumbleBee requires the target computer/network to run the software in some way or another. Phishing is one of the possible methods of achieving this, if the software never arrives at the target, it cannot be run.

## T1548.002 – Abuse Elevation Control Mechanism: Bypass User Account Control

When attacks occur on systems a lot of the software/scripts used may require administrator permission to run successfully. This can potentially stop an attack in its tracks if the user does not have that permission or simply clicks no if they do. This would be the case of BumbleBee, but to avoid this BumbleBee uses T1548.002. Without the use of this sub technique attacks would likely be stopped before anything can be run on the target machine.

## T1041 – Exfiltration Over C2 Channel

This technique is used to exfiltrate the data that has been stolen by attackers. The stolen data is returned to the attacker over the ‘C2’ channel. This is a communication channel which is usually hidden so systems cannot see it. This technique is important to BumbleBee as one of the primary goals of the software is to allow ransomware, so a channel for communication gives the attacker a way to retrieve the stolen information. If the software cannot communicate back the stolen data or send commands to run over the channel, then the software will not achieve its goal.

## T1053.005 – Scheduled Task/Job: Scheduled Task

The purpose of BumbleBee is to install and execute additional software and commands down the track after it has infected a system using WMI and the previous mentioned techniques. To achieve this some form of persistence needs to be secured. BumbleBee uses Scheduled Tasks to establish persistence. Without this sub technique no additional payloads could be executed or delivered after the initial delivery of BumbleBee.

# Attack Emulation

## Scripts

### T1047

“wmic process call create notepad.exe” example will open notepad.exe.

### T1053.005

“SCHTASKS /Create /SC ONCE /TN spawn /TR notepad.exe /ST 20:20” example will open notepad.exe at 20:20.

### T1548.002

“Invoke-AtomicTest T1548.002” can be used to emulate this technique.

## Detections/Safeguards

Wazuh has means of detecting each of these threats by creating new rules. The following rules will detect each of the 3 attacks.

<group name="windows,sysmon">

<rule id="115001" level="10">

<if\_group>windows</if\_group>

<field name="win.eventdata.ruleName"

type="pcre2">technique\_id=T1047</field>

<description>WMI execution on $(win.system.computer)</description>

<field name="win.eventdata.originalFileName" negate="yes" type="pcre2">.\*Wmiprvse.exe</field>

<field name="win.eventdata.originalFileName" negate="yes" type="pcre2">.\*wmiutils.dll</field>

<mitre>

<id>T1047</id>

</mitre>

</rule>

<rule id="115002" level="10">

<if\_group>windows</if\_group>

<field name="win.eventdata.ruleName"

type="pcre2">technique\_id=T1548</field>

<description>UAC bypass on $(win.system.computer)</description>

<mitre>

<id>T1548</id>

</mitre>

</rule>

<rule id="115003" level="10">

<if\_group>windows</if\_group>

<field name="win.eventdata.ruleName"

type="pcre2">technique\_id=T1053</field>

<description>New Schedualed task on $(win.system.computer)</description>

<mitre>

<id>T1053</id>

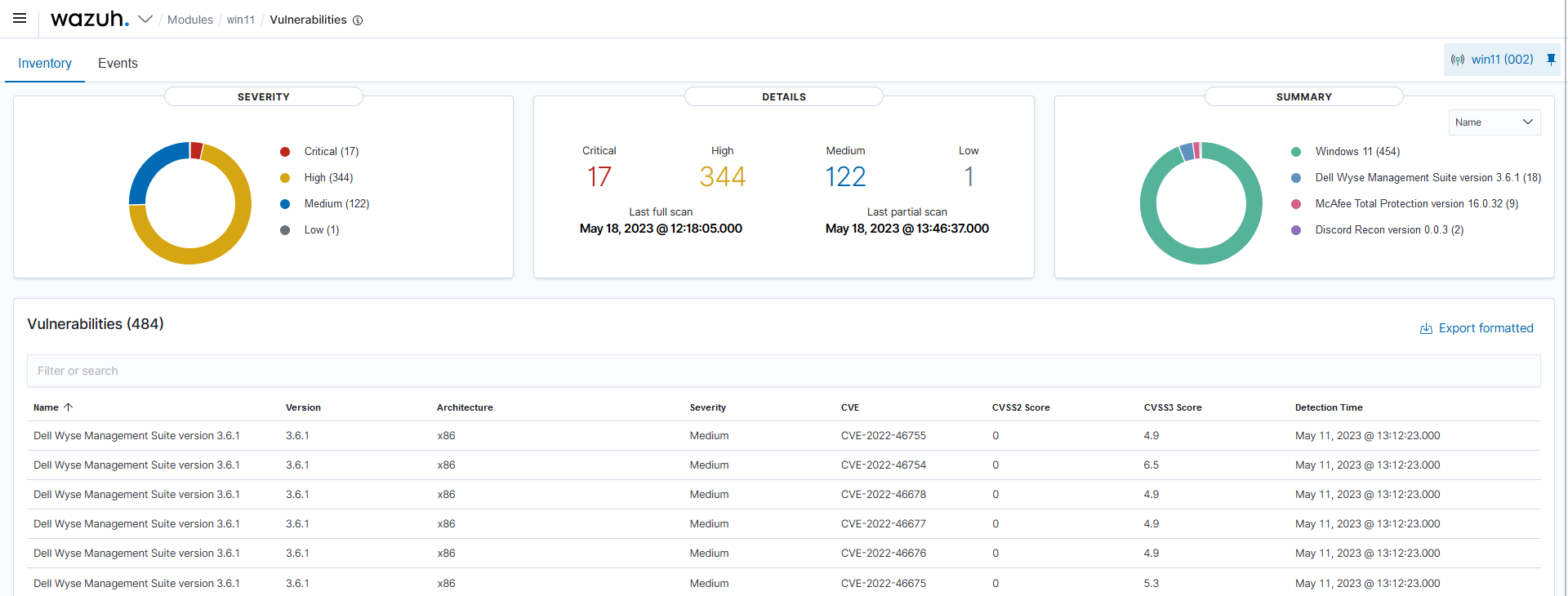
</mitre>

</rule>

</group>

# Vulnerability Search

## Wazuh



A screenshot of a computer

Description automatically generated with medium confidence

## Nessus

