



DISRUPT. DEFEND. UNITE.

Ransomware Variants Are Lurking “In the V-Shadows”

By Ryan Nolette
The man of many titles

Security Operations Lead
Senior Security Engineer
Senior Threat Researcher
Incident Response Consultant

©2016 Carbon Black. All Rights Reserved

CARBON
BLACK
ARM YOUR ENDPOINTS

Agenda

- **High level topics**
 - What is Ransomware?
 - What is vshadow?
 - How does an attacker abuse vshadow?
- **Visibility**
 - What happens on the host from the host point of view with CryptoLocker
- **Stopping this Ransomware threat**
 - How to prevent this infection with CryptoLocker V1



CARBON
BLACK
ARM YOUR ENDPOINTS

Hello everyone and welcome to my presentation. My name is Ryan Nolette and I currently run Security Operations for Carbon Black. As a disclaimer, I did use Carbon Black's products for some of the visuals in this presentation but I promise no one in marketing, or HR for that matter, has seen my presentation what I am going to show you today.

Today I would like to talk about how attackers are abusing a legitimate windows utility, called vshadow, to hold your data hostage and how to defend your enterprise from this threat.

I will give you a quick overview of who I am and what I do. Then I will explain at a high level what ransomware is and what these attacks look like on a system. After that, I am going walk you through a growing trend of abusing volume shadow copies on systems to disallow users from restoring from backups. Finally, I will end with ways you can quickly and easily detect and respond to these kinds of attacks. If possible, please hold your questions for the end of the presentation. I cut a bunch of information to make time and if I am unable to get to your question at the end, please find me later on and I will be more than happy to try and answer your additional questions.

The specific variant I will be detailing in this presentation is Cryptolocker version 1. I only have time during this presentation to go in depth for 1 variant and will be focusing on Cryptolocker V1.

<next slide>

What Can Ransomware Do?

Ransomware can:

- Prevent you from accessing Windows.
- Encrypt files so you can't use them.
- Stop certain apps from running (like your web browser).
- Demand that you do something to get access to your PC or files.
- Demand you pay money.
- **Make you complete surveys.**

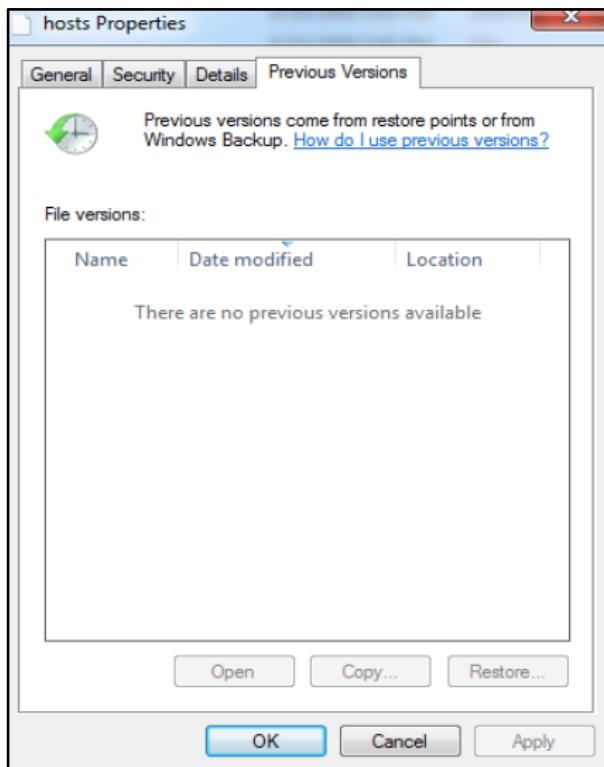


So what is Ransomware and what can it do? If you type “What is Ransomware” into Google, you get the definition of “a type of malicious software designed to block access to a computer system until a sum of money is paid.” That pretty well sums it up.

The options are pretty numerous to be honest. The most common options are encrypting your files and not allowing you to use your system without paying the ransom.

But by far the most devious and evil thing I have ever seen ransomware do *pause* *click*

Make you complete surveys *shudder*
<next slide>



What is vshadow?

VShadow is a command-line tool that you can use to create and manage volume shadow copies.

Also known as

- Shadow Copy
- Volume Snapshot Service
- Volume Shadow Copy Service
- VSS

CARBON
BLACK
ARM YOUR ENDPOINTS

VShadow is a command-line tool that you can use to create and manage volume shadow copies. Shadow Copy is a technology included in Microsoft Windows that allows the taking of backup copies of computer files or volumes. These backups can be taken even when the files are in use. It is implemented as a Windows service called the "Volume Shadow Copy Service" or "VSS".

What does that mean? This means the attackers now have a Microsoft signed binary to abuse.

I can see from some of your expressions that you can see why the removing of these files is beneficial to attackers. If you cannot recover from backups, you are at their mercy. <next slide>

Using Microsoft against itself with Volume Shadows

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\administrator>vssadmin list shadows
vssadmin 1.1 - Volume Shadow Copy Service administrator
(C) Copyright 2001-2012 Microsoft Corp.

No items found that satisfy the query.
```

A yellow callout bubble with the text "It looks like you're getting a virus" is positioned above the command prompt window. A cartoon illustration of a virus character is visible on the right side of the slide.

- We have seen the volume shadow service used for a number of things ranging from malware to penetration testing tools.

CARBON
BLACK
ARM YOUR ENDPOINTS

Some variants of the CryptoLocker ransomware family are known for deleting all volume shadow copies to prevent restoring from backup.

The ransomware does this by executing a delete shadows /all command.

I have observed various techniques utilizing volume shadows. Lately it has been utilized for avoiding detection and for anti-analysis.

The technique I am going to show you consists of:

1. attackers dropping their malware on the file system via whatever infection mechanism they choose
2. then create a volume shadow
3. "mount" the shadow and execute the malware
4. Then unmount and delete the shadow

What is unique about this technique is that even after the unmounting and deleting of the shadow, the executed malware will still run.

<click>

Now that we have laid some ground work, let's break something muuuuhahaha

<next slide>

Abusing Shadows

Create shadow

```
C:\>C:\Users\user\AppData\Local\Temp\vshadow.exe -p C:\  
USHADOW.EXE 3.0 - Volume Shadow Copy sample client.  
Copyright (C) 2005 Microsoft Corporation. All rights reserved.  
- Creation Time: 7/28/2015 12:31:02 PM  
- Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
```

Option Used
Shadow Name

Mounting the shadow with the “mklink” Command

```
Mklink Option  
mklink /D C:\Windows\System32\msdc \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
```

Directory of C:\Windows\System32\msdc			
07/16/2015	01:43 PM	<DIR>	iDefense
07/27/2015	10:07 AM		649,675 malware.exe

The malware is dropped into shadow and mounted
BLACK ARM YOUR ENDPOINTS

On Windows XP, the Vssadmin tool doesn't have the ability to create persistent shadows on the system. Starting with the Windows Vista SDK, Microsoft supplied a binary called Vshadow to allow this.

Once the Vshadow executable is on the victim, attackers can use it to create a persistent shadow. And by persistent, I mean survives between reboots. To create a persistent shadow attackers utilize the “-p” option and point it toward the location on the file system they want to create a shadow of.

In the above example, the attackers are creating a persistent shadow of the full C: drive. This will run for a few seconds and end with the output seen above.

Keep note of the “Shadow copy device name.”

(\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3) as it will be used to mount the shadow in the following attack

Now that the shadow with the malware has been created, it must be mounted. This is done using the “mklink” command.

Here the attackers are creating a symbolic link directory in System32 to a directory called “msdc.” The symlink directory points to the shadow copy of the C drive created earlier.

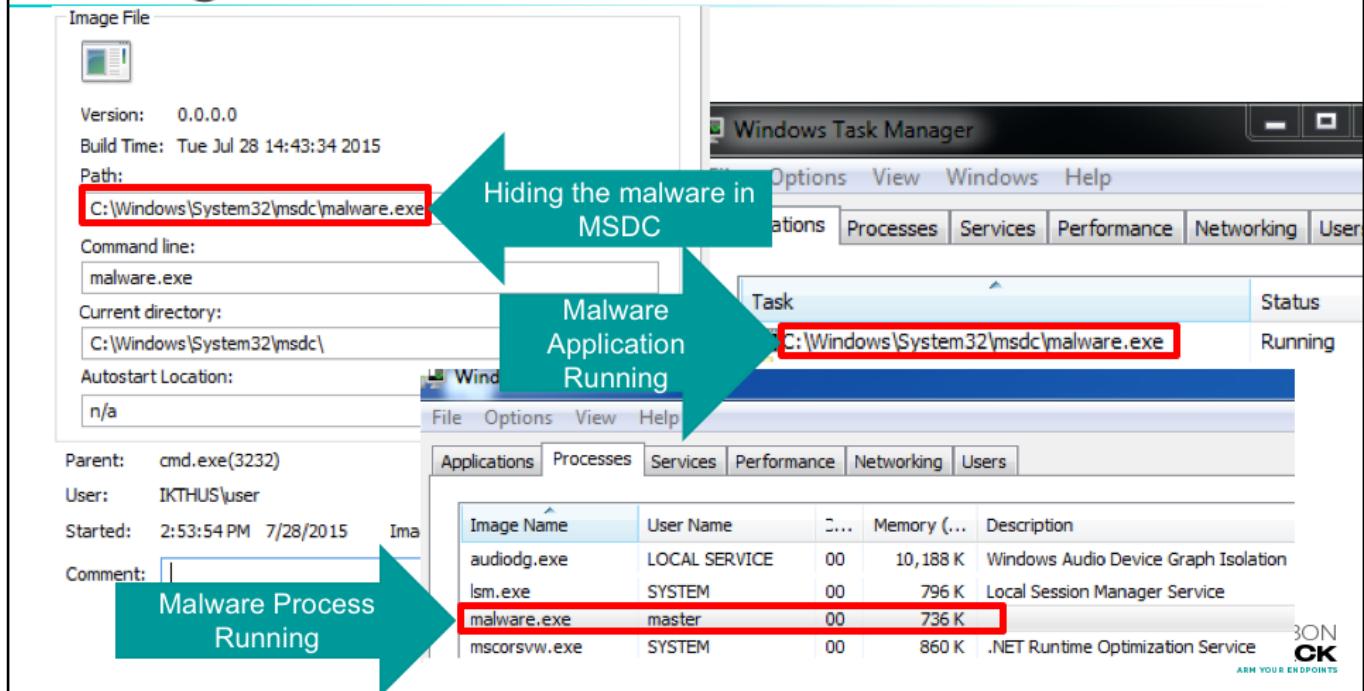
The malware was placed at the root of the shadow after it was created. A directory

listing of C:\Windows\System32\msdc reveals the malware on the normal filesystem but living inside the shadow filesystem.

Once the symlink has been created the contents of the shadow are accessible via normal file system operations like the directory listing we did above.

<next slide>

Starting the Malware



Once the file system setup is in place, the malware is started just like any other executable.

When the malware is started and shown in a tool like process explorer it shows that it is running from C:\Windows\System32\msdc

that path doesn't look too suspicious at first glance does it?
<next slide>

Malware Running After Shadow is Deleted

The screenshot shows a Windows terminal session with three windows:

- A command prompt window showing the path `C:\Windows\System32\msdc\malware.exe`. A red box highlights this path, and a teal arrow points from it to the text "Malware Process Running".
- A command prompt window showing the command `vssadmin delete shadows /all` being run. A red box highlights the command, and a teal arrow points from it to the text "Shadow Deleted".
- A Task Manager window showing two processes: `Administrator: C:\Windows\System32\cmd.exe` and `C:\Windows\System32\msdc\malware.exe`. Both processes are listed under the "Running" status column. A teal arrow points from this window to the text "Malware process still running after shadow is deleted".

Malware Process Running

Shadow Deleted

Malware process still running after shadow is deleted

Once the malware is started, the attackers can unmount and delete the shadow and the malware continues to run.

the attacker wants to remove as much forensic evidence as possible so they would unmount the directory and delete the shadow with Vssadmin

As demonstrated, this technique is a nice hiding mechanism that throws in a little anti-forensics with it.

<next slide>



This was my first hint that the infection worked. Unfortunately the AV installed on the demo system didn't catch it even though virustotal shows that my vendor had a signature for this malware.

Visibility is a key requirement of detection and preventions. I like to say, if you can't see it, how can you alert on it? And If you can't alert on it, how can you stop it?

Let's answer a few questions:

What really happened on the host?

What happens on the host from the host point of view?

<next slide>

What happens on the host from the host point of view

Search for all files created in last 30 days

```
Get-ChildItem -Path 'C:\' -Filter "*.exe" -Recurse | Where-Object { $_.CreationTime -gt (Get-Date).AddDays(-1) } | Select-Object Fullname,CreationTime | Out-File -FilePath c:\out.txt
```

```
Directory of C:\Users\master\AppData\Roaming
07/30/2015  04:00 PM    <DIR>      .
07/30/2015  04:00 PM    <DIR>      .
07/25/2015  12:22 AM    <DIR>      .
07/30/2015  04:00 PM      225,280 f1f94d81.exe
07/12/2015  10:02 AM    <DIR>      Ruemtices
04/12/2011  04:28 AM    <DIR>      Media Center I
      1 File(s)   225,280 bytes
      5 Dir(s)  43,191,943,168 bytes free

Directory of C:\Users\master\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
07/30/2015  04:00 PM    <DIR>      .
07/26/2015  04:00 PM    <DIR>      .
07/30/2015  04:00 PM      225,280 f1f94d81.exe
      1 File(s)   225,280 bytes
      2 Dir(s)  43,190,611,968 bytes free

Directory of C:\
07/14/2015  10:02 AM    <DIR>      $Recycle.Bin
07/15/2015  10:02 AM    <DIR>      Boot
11/20/2010  11:23 PM      383,786 bootmgr
07/14/2015  01:58 PM      8,192 BOOTSECT.BAK
07/14/2015  01:58 PM    /JUNCTION/ Documents and
07/30/2015  04:00 PM    <DIR>      f1f94d81
07/27/2015  04:00 PM    /JUNCTION/ Userer.DAT.SUS

Directory of c:\f1f94d81
07/30/2015  04:00 PM      225,280 f1f94d81.exe
      1 File(s)   225,280 bytes
      0 Dir(s)  43,187,834,880 bytes free
```

The first thing this malware does is delete itself from the original location it was executed from and create a new binary in the user's appdata roaming directory.

How do I know this? Because on the suspected compromised computer I ran a powershell query to find all new files created in the past 24 hours.

From this list I was able to quickly find randomly generated executable files in strange directories like appdata.

This is extremely common among Trojan malwares and is the first place I check for newly created directories and binaries because it is so common.

Next the malware creates a persistence mechanism by copying itself to the user's startup programs directory.

This is a common technique and is a location that should always be checked for new binaries.

The third action this malware takes is to create a hidden folder in the root directory of the files system.

You can see that the folder was created within seconds of the original binary being deleted and the other two binaries being written to the filesystem.

Inside this new file is yet another copy of the binary.

It seems like the malware author is afraid of these binaries being found and creates backup plans for their backup plans.

That kind of paranoia isn't healthy.

<next slide>

Finding the application Hash

```
certUtil -hashfile pathToFileToCheck HashAlgorithm
```

```
HashAlgorithm choices: MD2 MD4 MD5 SHA1 SHA256 SHA384 SHA512
```

```
C:\>cd Windows\System32  
C:\Windows\System32> certUtil -hashfile cmd.exe MD5  
MD5 hash of file cmd.exe:  
f5 ae 03 de 0a d6 0f 5b 17 b8 2f 2c d6 84 02 fe  
CertUtil: -hashfile command completed successfully.  
  
C:\Windows\System32> certUtil -hashfile cmd.exe SHA256  
SHA256 hash of file cmd.exe:  
6f 88 fb 88 ff b0 f1 d5 46 5c 28 26 e5 b4 f5 23 59 8b 1b 83 78 37 7c 83 78 ff eb  
c1 71 ba d1 8b  
CertUtil: -hashfile command completed successfully.
```



SHA256: eafe38f481344f23bb9d783fc21c734b2cd37d4a3f37e4a5a282fd739a87316b

File name: d0bfc139.vxe

Detection ratio: 44 / 56

Analysis date: 2015-08-01 10:57:35 UTC (9 months, 2 weeks ago)



CARBON
BLACK
ARM YOUR ENDPOINTS

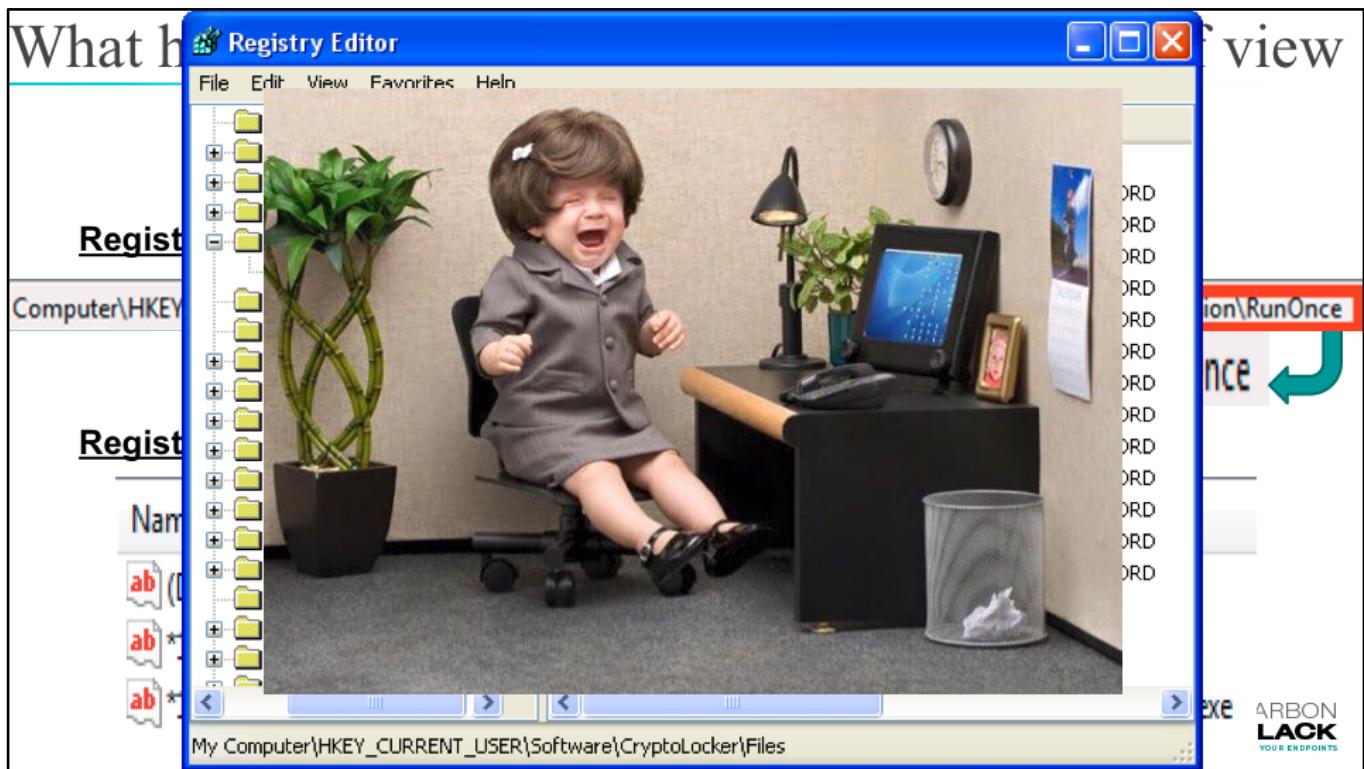
While in the process of researching different ways to hash files, I discovered that windows has a native hashing feature built into certUtil.

Using this tool I can programmatically hash every file on a system and then upload that hash list to virustotal for a quick check for known malware.

When I did this for the binary, I found that it has a virustotal score of 44/56.

We should probably look into what this binary did right?

<next slide>



Next up, the malware starts creating registry values so it can be started in the background each time the user logs in.

I can infer their intent because the “Run” and “RunOnce” keys are run each time a new user logs in and would start this malware again upon login.

These keys are for background services such as remote registry service and are run only once per boot.

This is what I have been referring to as the “pissed list” because the longer the list, the more you're pissed.

<next slide>

How to detect this attack

- IOC's
 - Hashes
 - Filenames/paths
 - Registry Values
 - Network Connections
- Behaviors
 - Loading of Dependencies
 - Process of Execution
 - Usage of rarely executed native tools

Hashes

- MD5 (vshadow-7-32.exe) = 3e1360a23ea5f9caf4987ccf35f2fcfa
- MD5 (vshadow-7-64.exe) = 576b379a59d094fb7b06c261a96034a6
- MD5 (vshadow-8-32.exe) = d0cd7ad91b2ff568275d497214ff185c
- MD5 (vshadow-8-64.exe) = 97fd0f3c05f1707544a9a6a0c896b43e
- MD5 (vshadow-8.1-32.exe) = d560c155b68121d98f8370e7deafbc4d
- MD5 (vshadow-8.1-64.exe) = c5d2992c8cba0771f71fe4d7625a0b8b
- MD5 (vshadow-vista-64.exe) = 53d3e33ad31af6716559f29e889aca49

I
N
D
I
C
A
T
O
R
S

CARBON
BLACK
ARM YOUR ENDPOINTS

Next question is “how do I detect this type of attack?”

I tried to stick to indicators that could be used in various tools.

Our first method for finding use of the Vshadow tool is looking for hashes. Each version of the SDK will have the Vshadow tool in it and will have an x86 and 64bit version.

<next slide>

Finding Vshadow Being Used

- Detect loading of DLL and ignore werfault
 - modload:vss_ps.dll cmdline:"-p" -path:System32\werfault.exe
- Command line or batch file usage fo mklink
 - cmdline:""C:\Windows\system32\cmd.exe" /c mklink /D"
- Look for vshadow being run
 - process_name:vshadow.exe AND cmdline:"-p C:\"

```
modload | Loaded c:\windows\system32\vss_ps.dll Signed (4d4e2a2fe9c824733c7a53f2e5454aff)
```

- path:device/harddiskvolumeshadowcopy*
- path:device/harddiskvolume*

Hiding the malware in
Shadow

Process: malware.exe

PID: 2680

OS Type: windows

Path: \device\harddiskvolumeshadowcopy3\malware.exe

Username: IKTHUS\user

MD5: 1f04721b1cea854077288fcf5d91f96f

Start Time: 2015-07-28T18:36:30.276Z

If we look a little closer at the vshadow.exe process we can see it loads a few modules that don't normally get loaded. One in particular that we are looking for is vss_ps.dll, which is a necessary component of the Volume Shadow Storage feature. I figured this out by googling vshadow.exe and the MSDN site listed it as a dependency.

In a 3,000-host environment, that query came back with only one process that matches the criteria. The process is the Windows process werfault.exe. So we can refine the query to ignore this process.

One caveat I found while researching this is the command "mklink" is a function of cmd.exe. Because of this, it is going to be hard for some IR tools to detect. luckily the tool I was using can see the command line and can detect this as written in the fourth query

When taking a closer look at the malware.exe process we can see that the true file system path is \Device\harddiskvolumeshadowcopy3\malware.exe

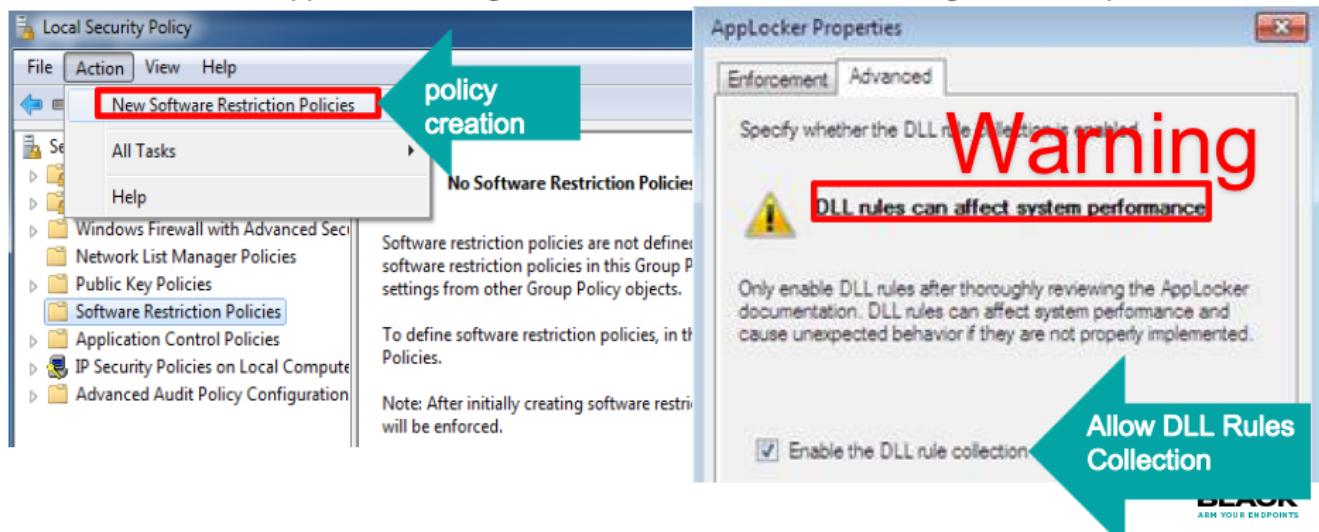
This means we can flag this malware being executed from the volume shadow, along with other processes being run from locations that have "device/harddiskvolume" in the path.

<next slide>

How to prevent this infection

The file paths that have been used by this infection and its droppers are:

- C:\f1f94d81\f1f94d81.exe
- C:\Users\master\AppData\Roaming\f1f94d81.exe
- C:\Users\master\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\f1f94d81.exe



You can create a software restriction policy for a single computer using the Local Security Policy Editor or for an entire domain use the Group Policy Editor

Fun Fact: you can use applocker to enforce DLL's as well as binaries.

Second Fun Fact: stick to just binaries if you can because this will degrade system performance and that makes users very sad pandas.

The DLL Rules Collection

The DLL rules collection is used to block applications that call specific DLL files.

This is an advanced rule collection and should not be used unless you are certain you know what you are doing.

This type of rule can also severely impact system performance as it requires AppLocker to check every DLL an application uses when it initializes.

The DLL rules collection is not enabled by default due to the reasons mentioned above. If you want to create a DLL rule you can do so by going to the main AppLocker configuration screen, choosing Configure Rule Enforcement, selecting the Advanced tab, and placing a check mark next to the Enable the DLL rule collection option. After doing this you will see the DLL rule collection in the left pane alone with the three other rule collections.

<next slide>

How to prevent this infection

The screenshot shows a comparison between a list of security rules and a configuration dialog box.

List of Rules:

- Block executable in %AppData%**
 - Path: %AppData%*.exe
 - Security Level: Disallowed
- Block executable in %LocalAppData%**
 - Path if using Windows XP:
%UserProfile%\Local Settings*.exe
 - Path if using Windows Vista/7/8:
%LocalAppData%*.exe
 - Security Level: Disallowed
 - Description: Don't allow executables to run from %AppData%

New Path Rule Dialog Box:

- General Tab:** Use rules to override the default security level.
- Path:** %AppData%*.exe
- Security level:** Disallowed
- Description:** Block executable in %AppData%

BLAUK ARM YOUR ENDPOINTS

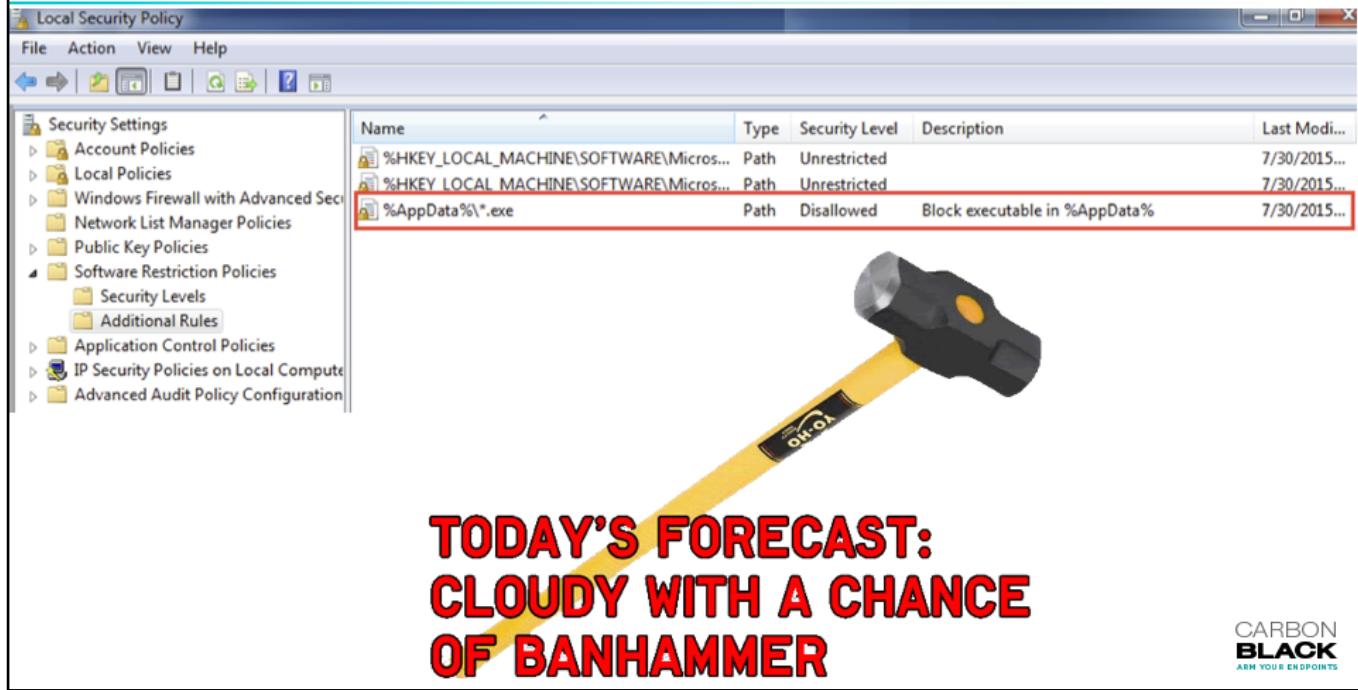
The software restriction policies can cause issues when trying to run legitimate applications from weird locations. For them, you will need to add exception rules.

There is some trial and error here since this enforcement technique is the equivalent of trying to open a jar of pickles with a sledgehammer.

Applications like Chrome and Spotify are known to use AppData or a child directory of appdata for update binaries. So beware of collateral damage.

<next slide>

How to prevent this infection



The screenshot shows the Windows Local Security Policy snap-in. The left pane displays a tree view of security settings, including Account Policies, Local Policies, Windows Firewall with Advanced Security, Public Key Policies, Software Restriction Policies, Application Control Policies, IP Security Policies on Local Computer, and Advanced Audit Policy Configuration. The right pane lists Software Restriction Policies. A policy named "%AppData%*.exe" is selected and highlighted with a red border. This policy has a "Type" of "Path", a "Security Level" of "Unrestricted", and a "Description" of "Block executable in %AppData%". The date "7/30/2015..." is listed under "Last Modified" for both the policy and its description. A large yellow hammer is overlaid on the right side of the slide, pointing towards the selected policy. Below the hammer, the text "TODAY'S FORECAST: CLOUDY WITH A CHANCE OF BANHAMMER" is displayed in large red letters. In the bottom right corner, there is a logo for Carbon Black with the tagline "ARM YOUR ENDPOINTS".

Name	Type	Security Level	Description	Last Modif...
%HKEY_LOCAL_MACHINE\SOFTWARE\Micros...	Path	Unrestricted	7/30/2015...	
%HKEY LOCAL MACHINE\SOFTWARE\Micros...	Path	Unrestricted	7/30/2015...	
%AppData%*.exe	Path	Disallowed	Block executable in %AppData%	7/30/2015...

**TODAY'S FORECAST:
CLOUDY WITH A CHANCE
OF BANHAMMER**

CARBON
BLACK
ARM YOUR ENDPOINTS

Save the rule and drop that ban hammer

<next slide>

Flag it, Tag it, and Bag it.

Ryan Nolette
Carbon Black

<https://github.com/sonofagl1tch>

<https://www.carbonblack.com/author/ryan-nolette/>

DISRUPT. DEFEND. UNITE.

CARBON
BLACK
ARM YOUR ENDPOINTS

In closing, Ransomware is annoyingly effective.

The recent additions of features such as removing shadow copies makes it even more dangerous.

Regardless of what security products you use, your best defense to any attack is user training and backups.

Anything preventative you can implement proactively, whether it's automated tools or a manual implementation is going to help protect you and your company.

Thank you all for your time today and until next time, remember my motto: "Flag it, Tag it and Bag it."

<next slide>