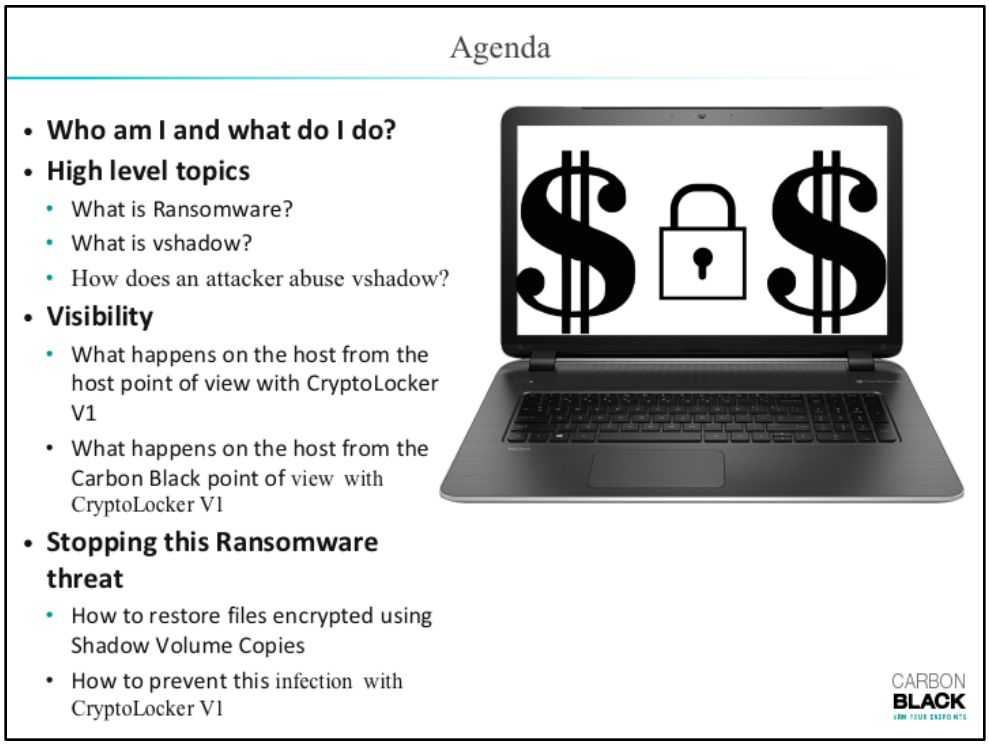
Ransomware Variants Are Lurking "In the V-Shadows"

By Ryan Nolette
Security Operations Lead
- Senior Security Engineer
- Senior Threat Researcher
- Incident Response Consultant

Carbon Black

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 this 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.

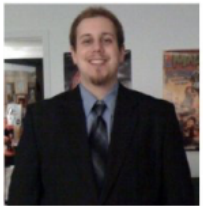
Today I would like to talk about how attackers are abusing a legitimate windows utility, called vshadow, to hold your data hostage and how to defend your enterprise from this threat.

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>

$ whoami



- **My name is Ryan Nolette**
  - I am currently the **Security Operations Lead** at Carbon Black
    - **Manage Security Operations**
    - **Act as Senior Security Architect for Carbon Black**
  - 10+ year veteran of IT, Incident Response, Threat Research, and Forensics
  - Carbon Black blog link
    - https://blog.CarbonBlack.com/author/rnolette/
  - **Responsibilities**:
    - Monitor Endpoint Events, Network Based Events, and Physical Security Events
    - User Education and Outreach
    - IT Oversight and Assistance
    - Security Oversight of Enterprise Projects
    - Incident Response
    - System Forensics
    - Vulnerability Scanning
    - Threat Research
    - ETC

CARBON BLACK

So who am I and why should you bother to pay attention to what I am saying?

I currently run Security Operations at Carbon Black and concentrate on the day to day SOC operations and security solutions required by the business.

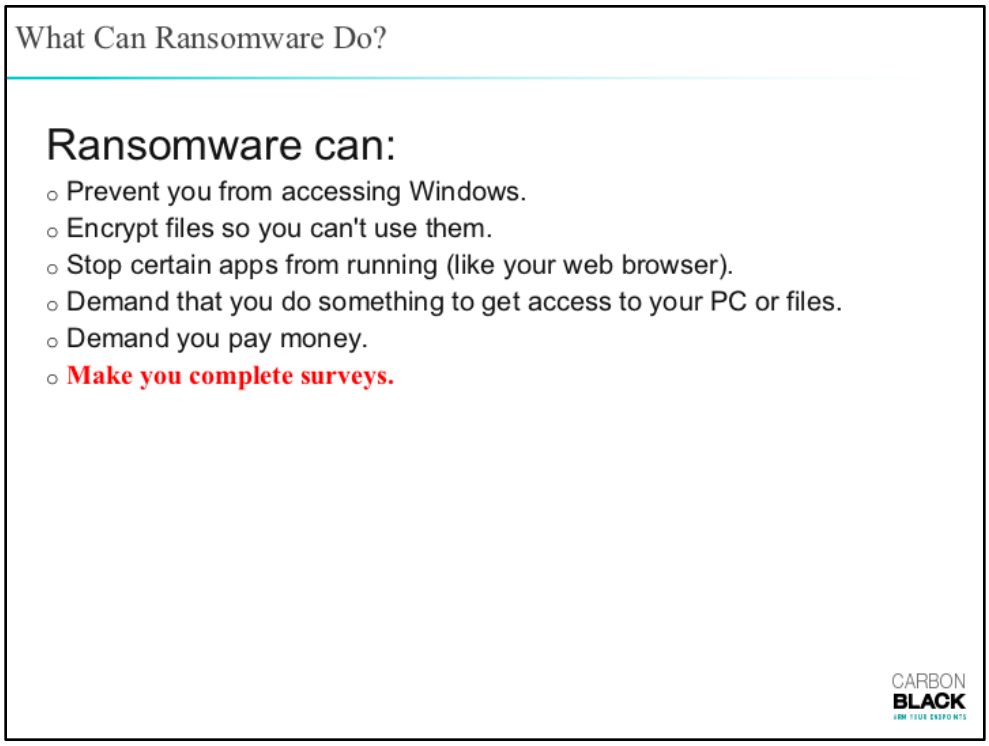
Before I took over SecOps I was a Senior Threat Researcher for Carbon Black and Senior Incident Response Consultant for external clients of Carbon Black.

You can see the long list of high level responsibilities but It can all be summed up pretty quickly... <next slide>

basically, I make sure that not only do we eat our own dogfood but that we also clean the messes left behind

<wait for laughs to finish then next slide>

So what is Ransomeware? 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.
<next slide>



What can Rasomware do?

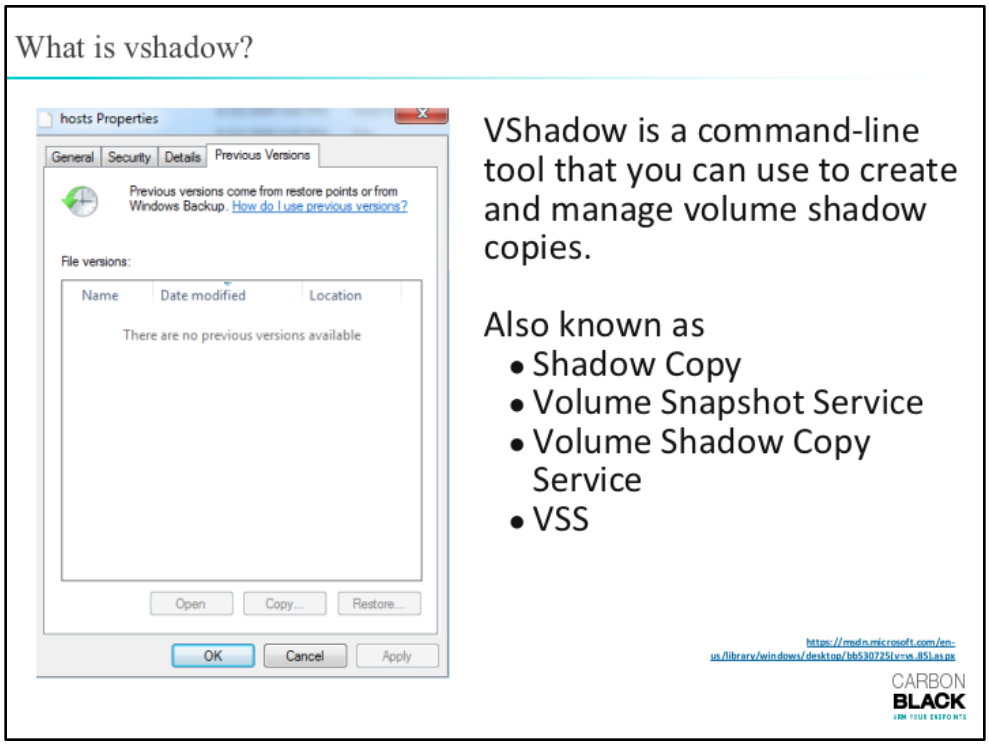
The options are pretty numerous to be honest. The most common options are encrypting your files and not allowing you to use your system without paying the ransom.

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?

<next slide>

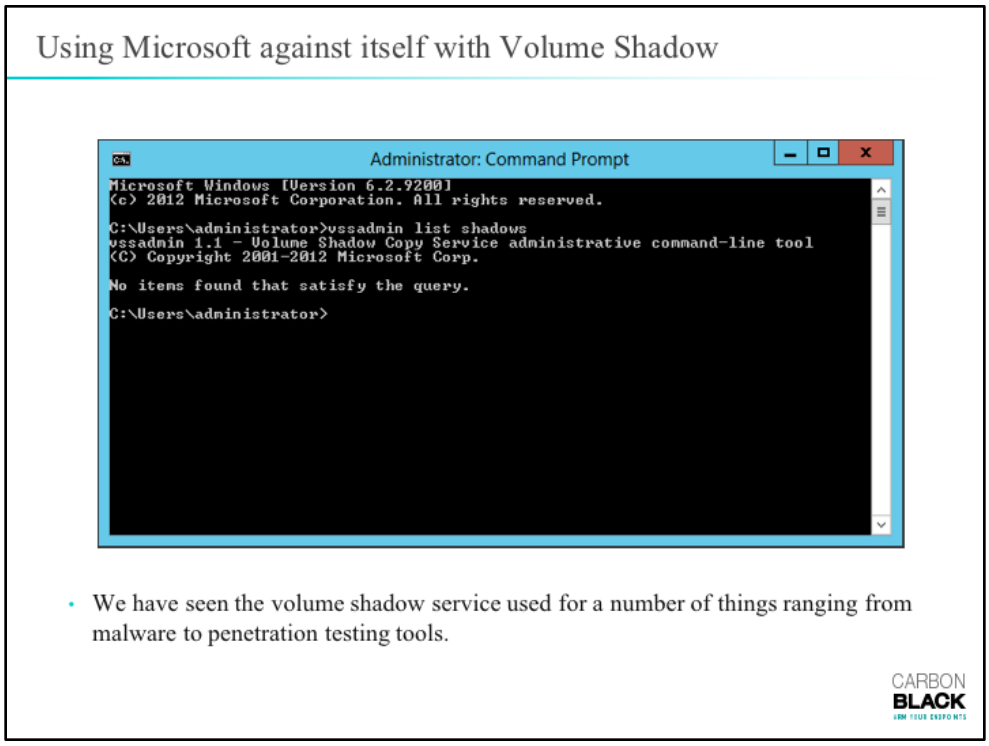


VShadow is a command-line tool that you can use to create and manage volume shadow copies. Shadow Copy is a technology included in Microsoft Windows that allows the taking of backup copies of computer files or volumes. These backups can be taken even when the files are in use. It is implemented as a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

Why is this significant? 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>

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

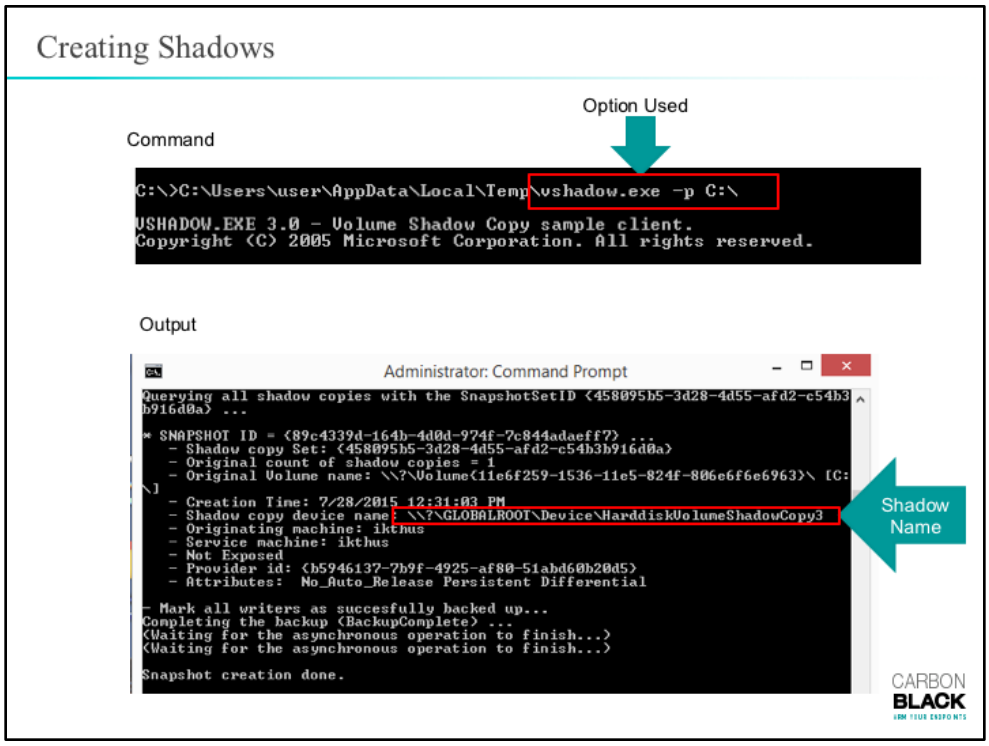


Some variants of the CryptoLocker ransomware family are known for deleting all volume shadow copies to prevent restoring from backup.
The ransomware does this by executing a delete shadows /all command.

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

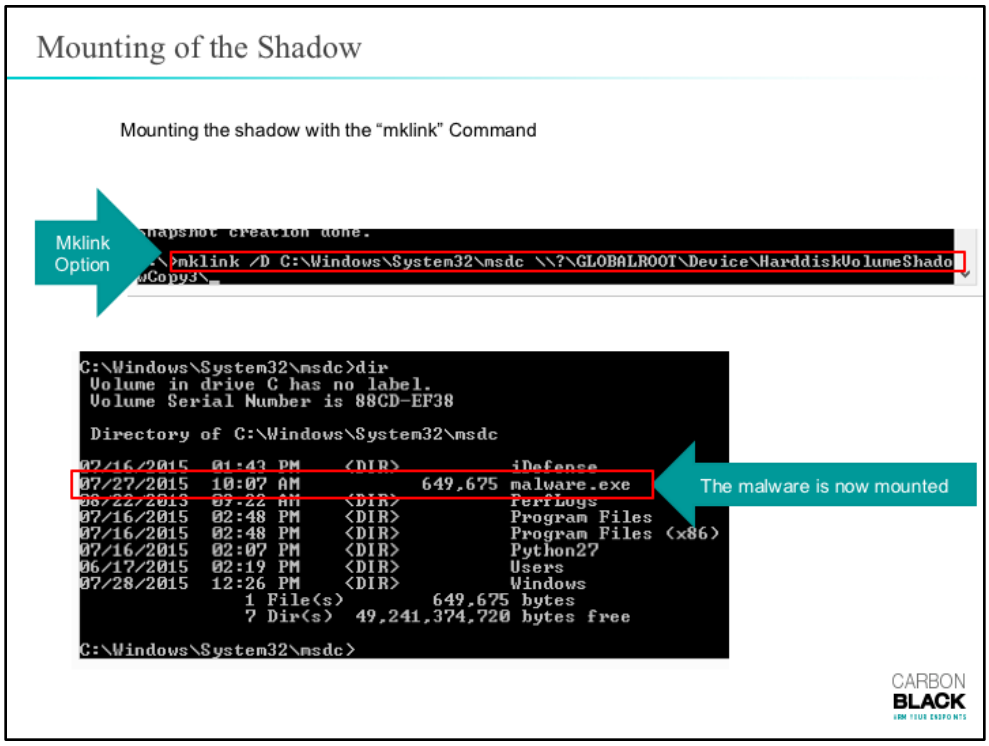
What is unique about this technique is that even after the unmounting and deleting of the shadow, the executed malware will still run.
<next slide>



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
<next slide>

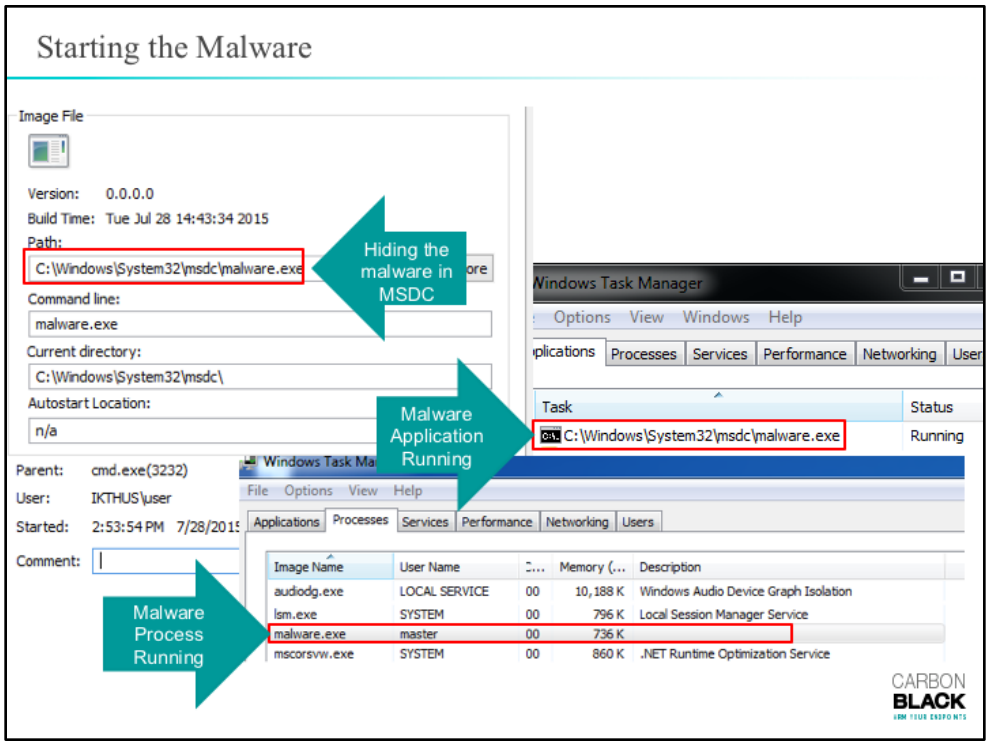


Now that the shadow with the malware has been created, it must be mounted. This is done using the "mklink" command.

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 is 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.

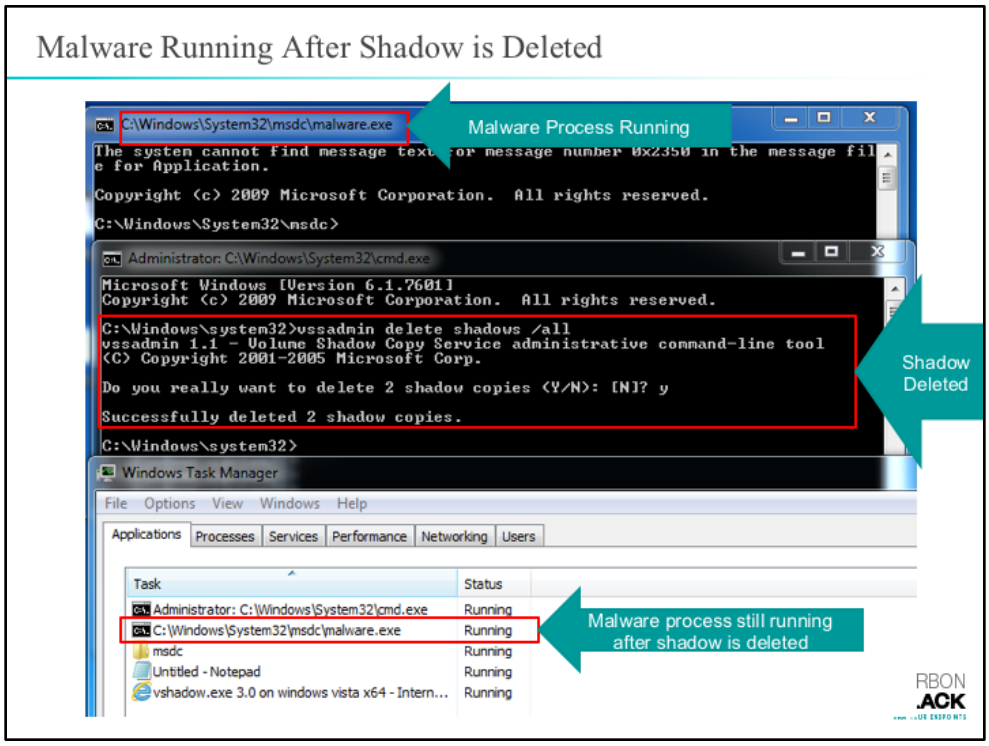
Once the symlink has been created the contents of the shadow are accessible via normal file system operations like the directory listing we did above.
<next slide>



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 doesnt look too suspicious at first glance does it?
<next slide>



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

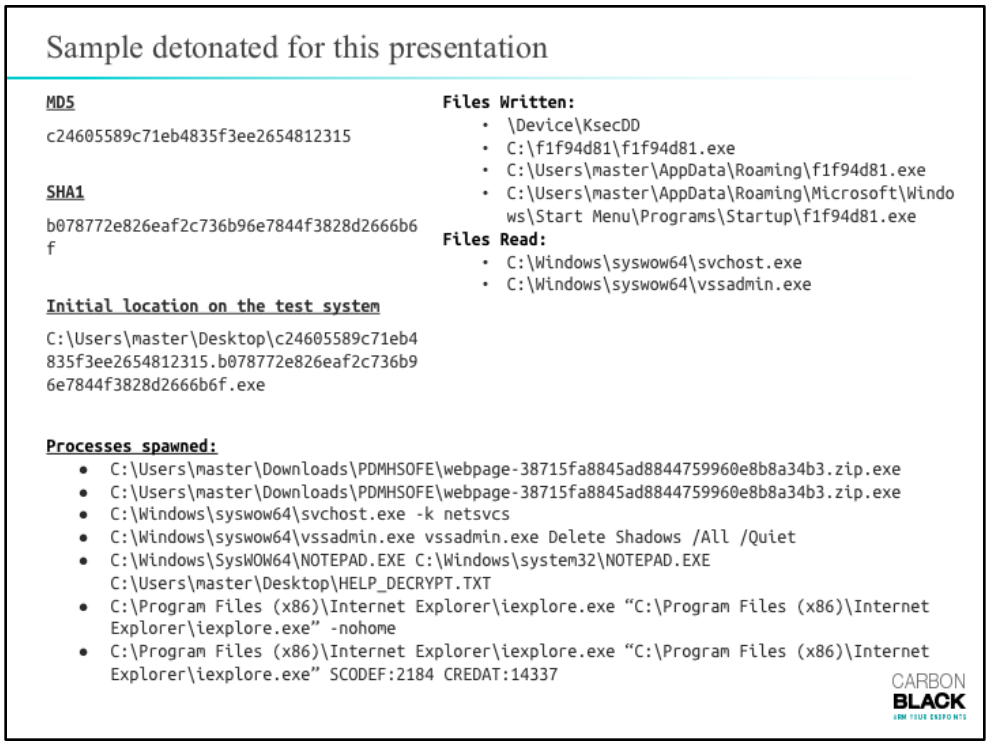
the attacker wants to remove as much forensic evidence as possible so they would unmount the directory and delete the shadow with Vssadmin

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

Now that we have gone over a high level explanation of Ransomeware and the recent examples of abusing vshadow, let's talk about visibility.

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 happens on the host from the host point of view?
What happens on the host from my IR tools' point of view?
<next slide>



These are the details about the sample I detonated for the rest of the presentation.

No you do not have to memorize this slide…..but there will be a test later…

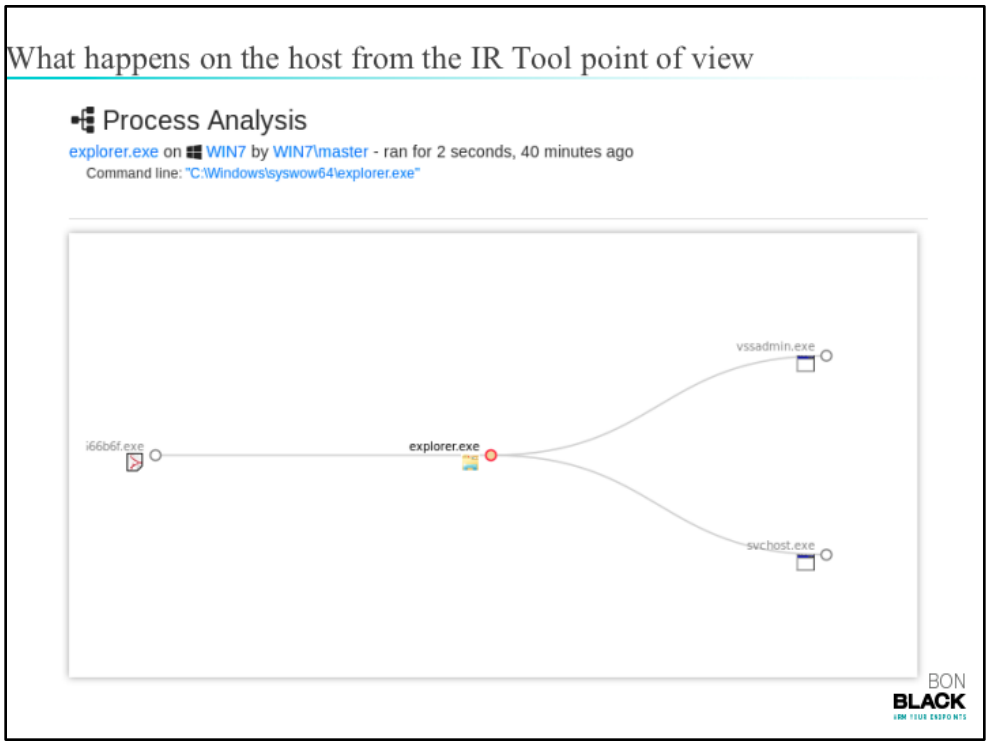
Since we detonated this malware on purpose, we already know what to look for based on the original filename. In the real world, we don't have that luxury.

To make this workflow as realistic as possible I created a new watchlist on my Carbon Black server looking for vshadow being run then detonated the malware.

These conditions would occur if vssadmin was executed from command line or from a batch script.

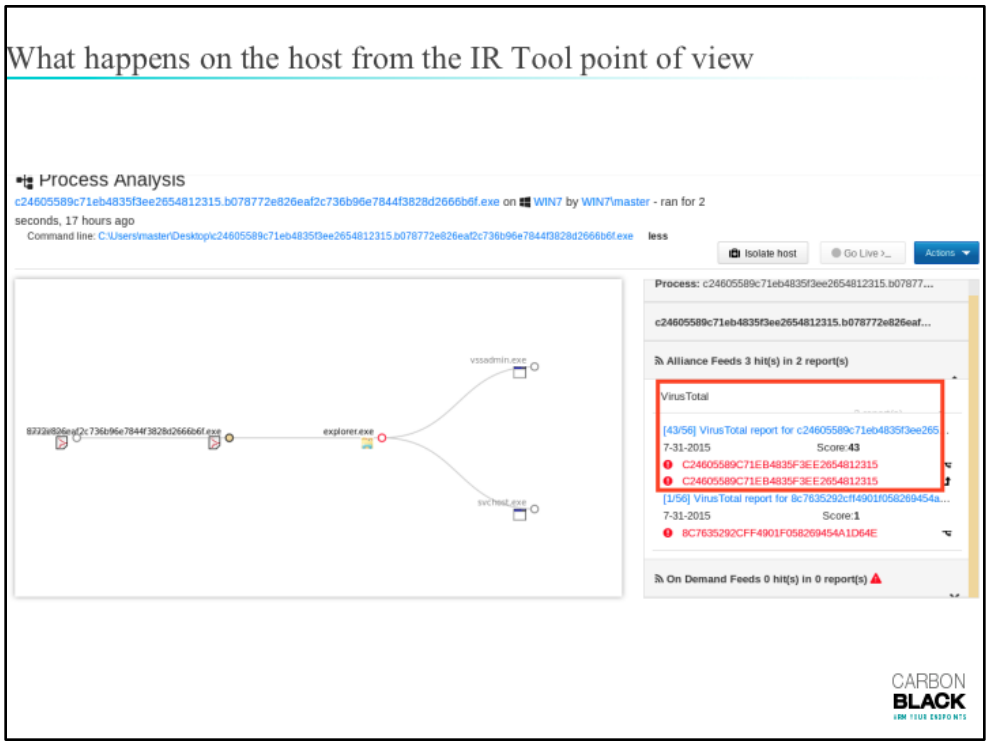
<next slide>

What happens on the host from my IR tools' point of view?
<next slide>



In our initial investigation, we can see the process tree of this file. Based on the alert, which led me to the vssadmin.exe execution, I was able to backtrack up the tree to see the originating file and process

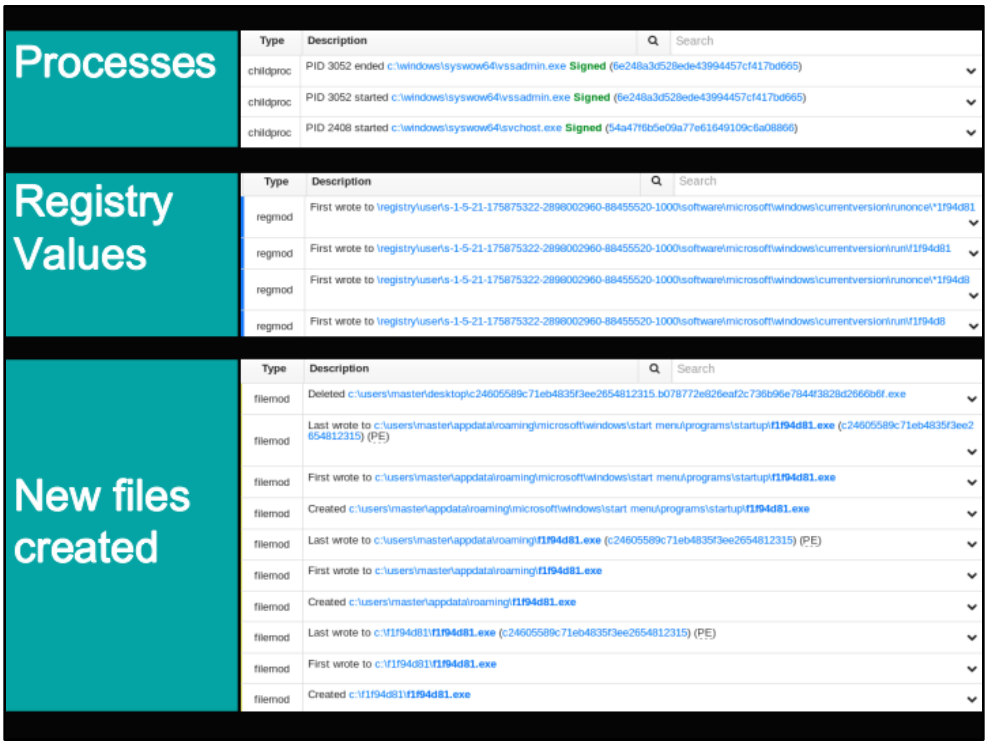
<next slide>



Looking into more details about this binary, we can see that the originating process has a virustotal score of 43/56.

We should probably look into what this binary did.

<next slide>

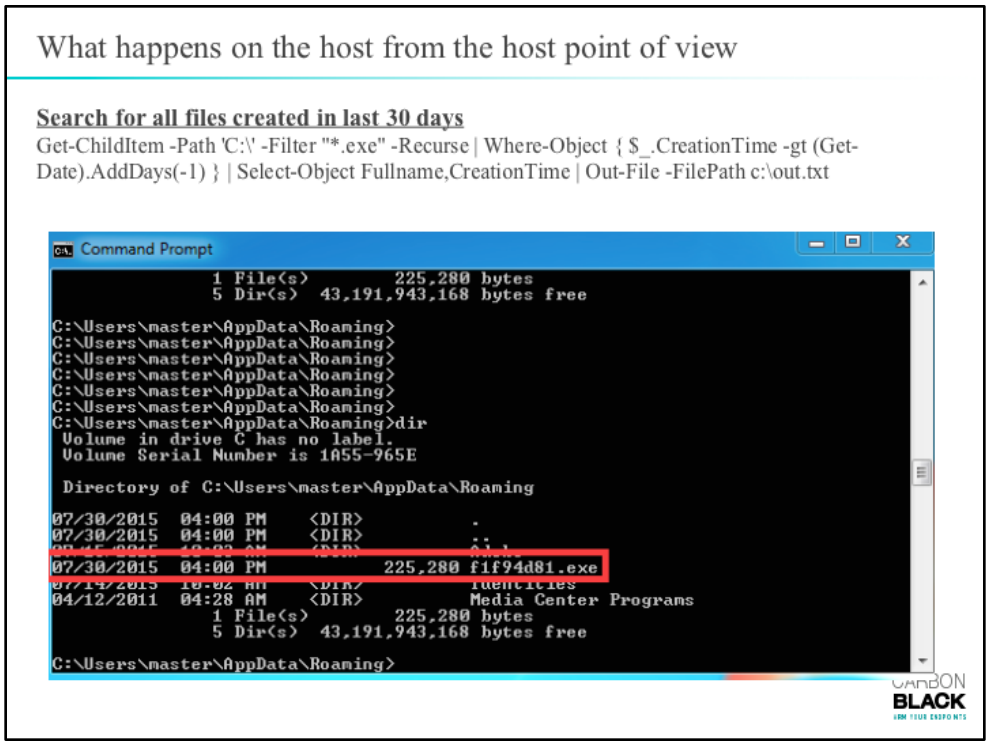


Above, we can see the behind the scenes actions of the malware. It spawned three new processes, created four new registry entries, and created 10 new files on the system.

Based on our findings, we can assume we're owned and need reimaging

Or the more manual approach of kill the active processes, remove all the files and registry entries that were created, and then restore the files from backups.
<next slide>

What happens on the host from the host point of view?
<next slide>



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.

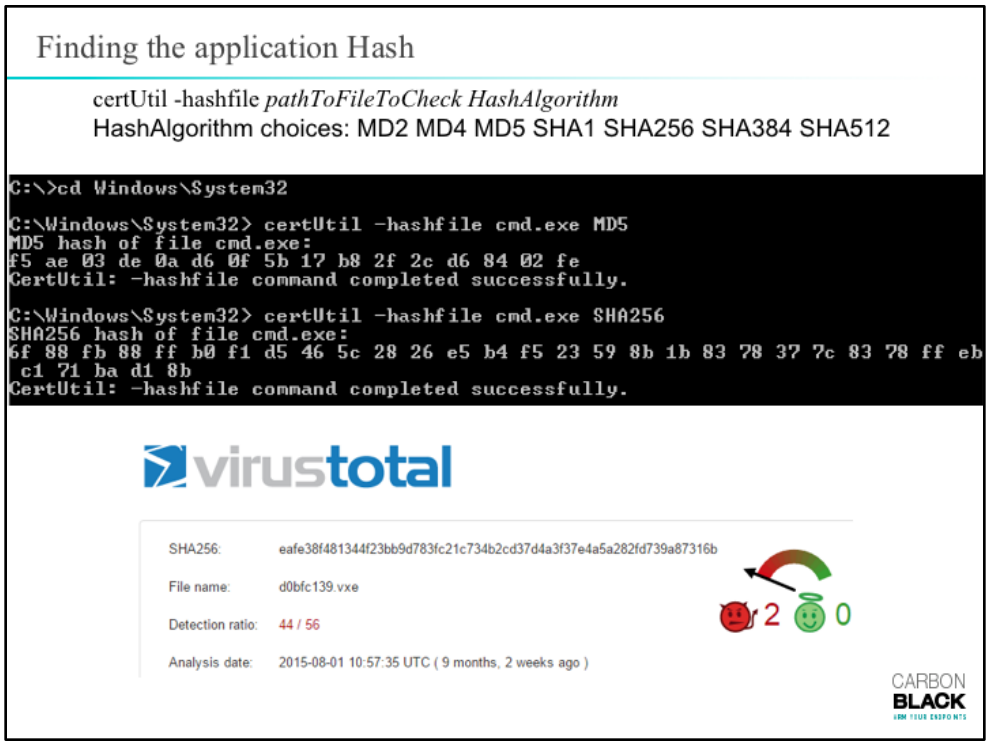
How do I know this? Because on the suspected compromised computer I ran a powershell query to find all new files created in the past 24 hours.

From this list I was able to quickly find files with randomly generated executable names in 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.

Now that we found a thread to pull, let's pull it and see where it leads us.

<next slide>



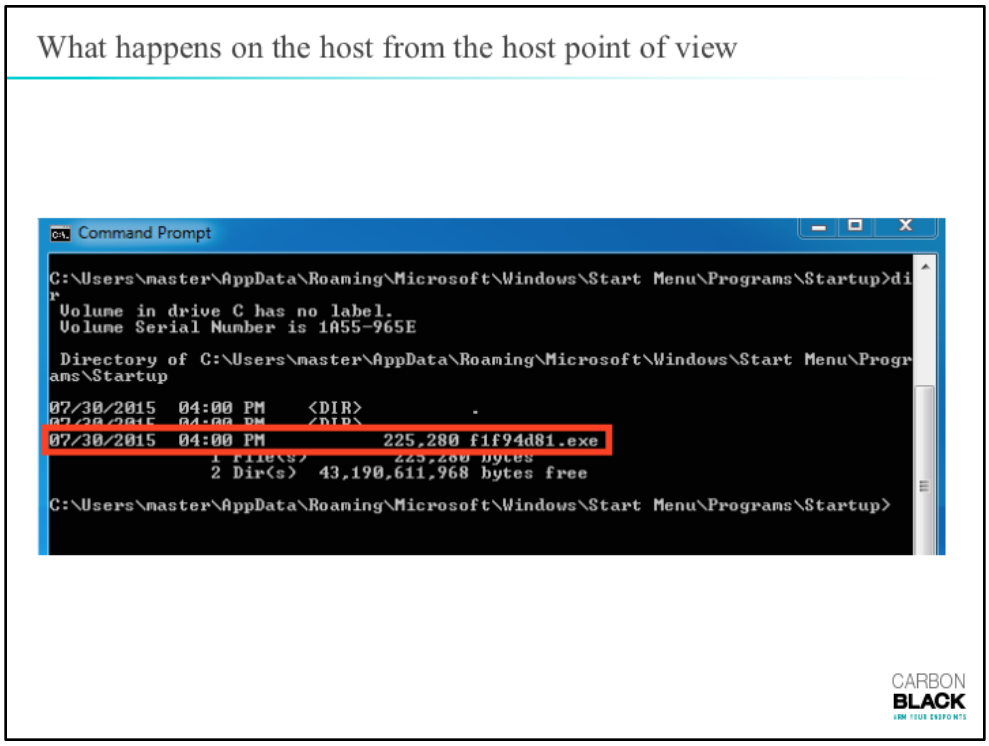
While in the process of researching different ways to hash files, I discovered that windows has a native hashing feature built into certUtil.

Using this utility 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 43/56.

We should probably look into what this binary did right?

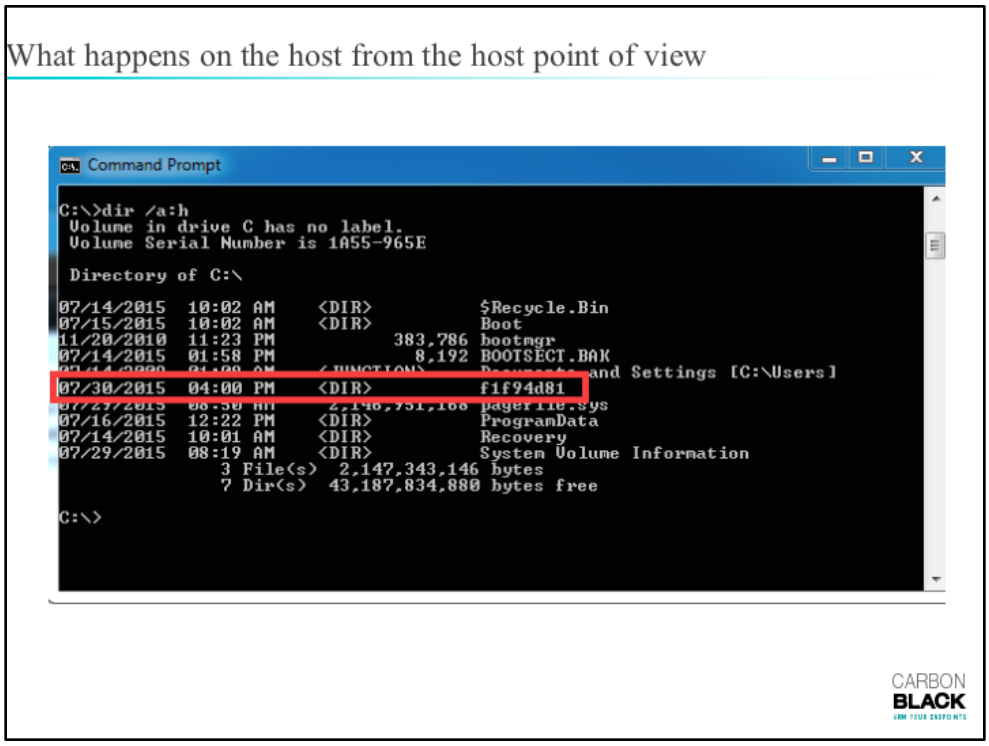
<next slide>



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

This is a common technique for malware trying to hide itself and is a location that should always be checked for new binaries.

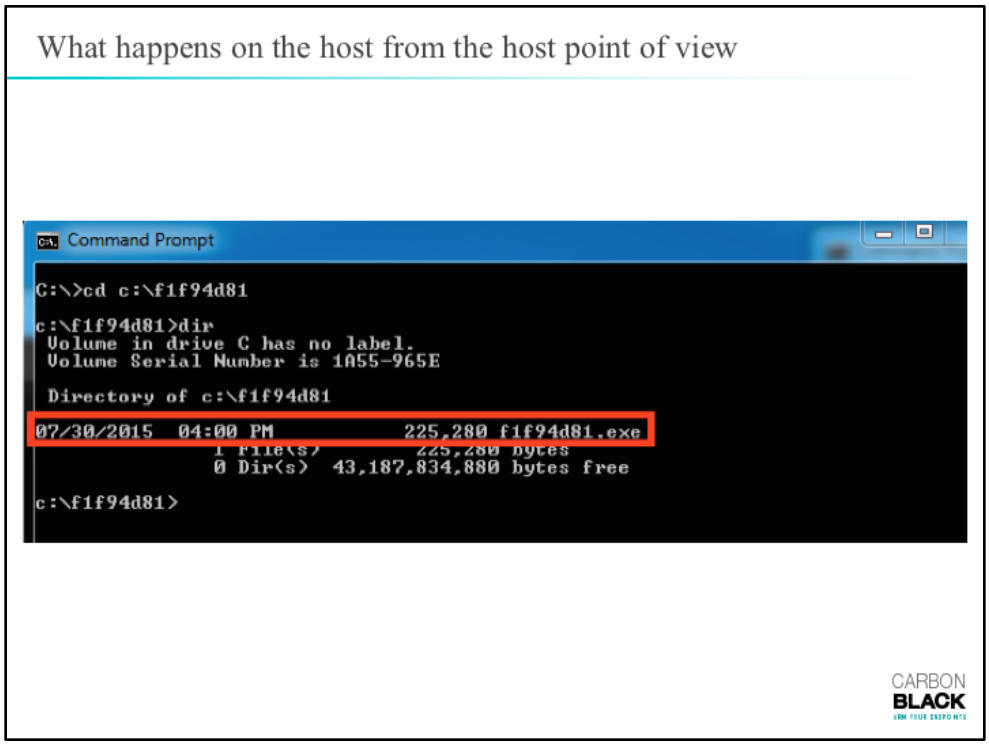
<next slide>



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.

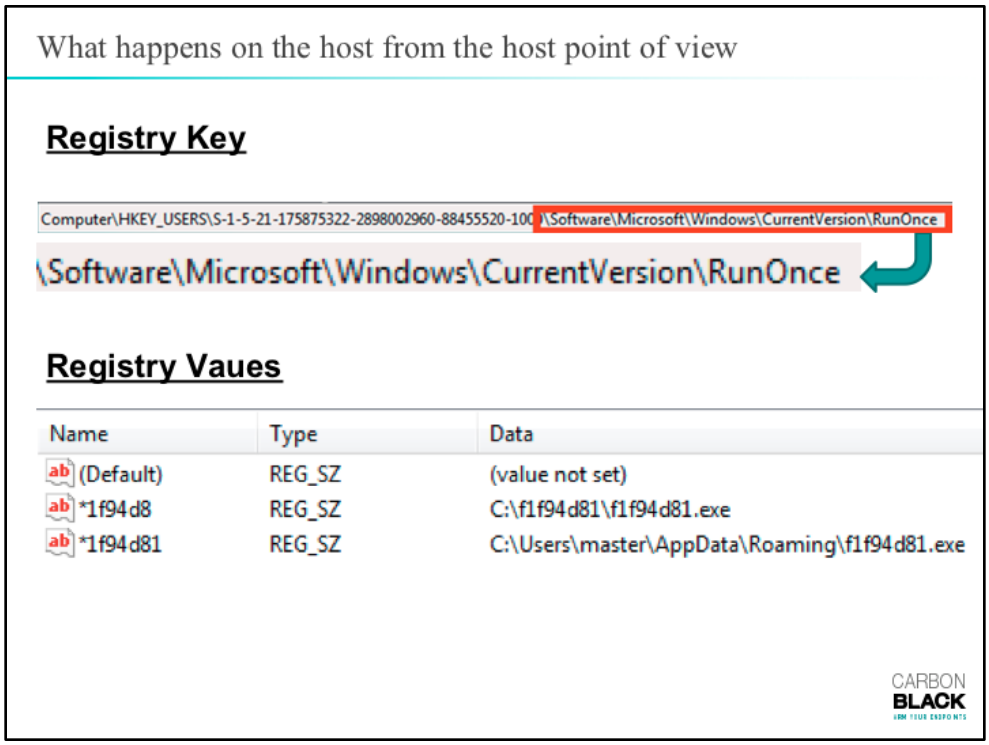
<next slide>



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>



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.

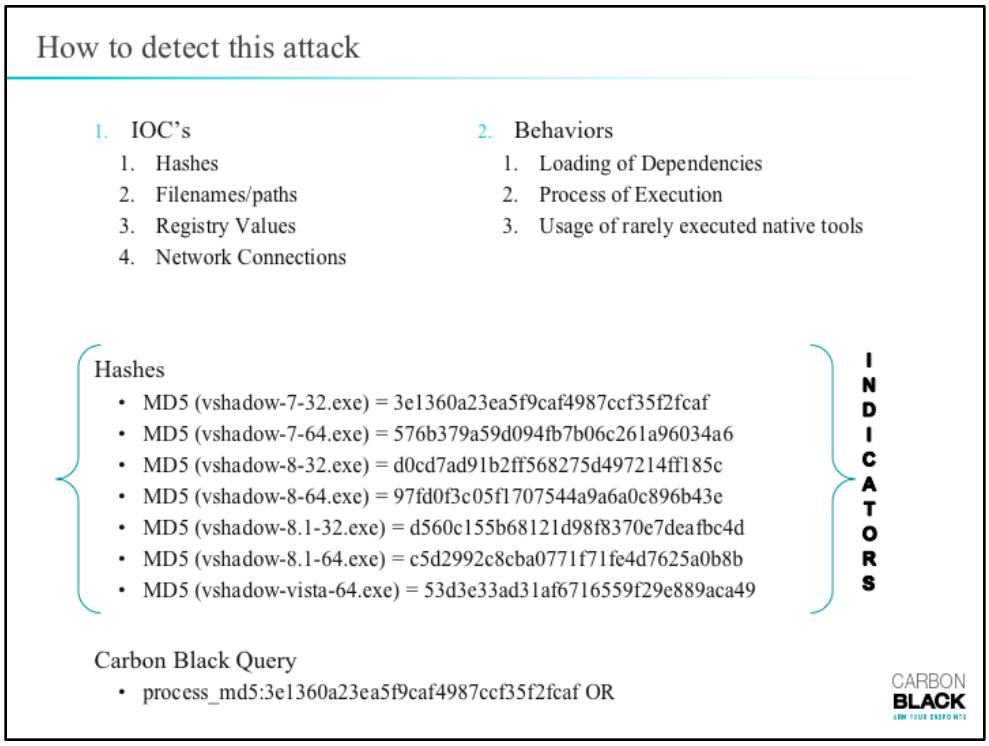
These keys are for background services such as remote registry service and are run only once per boot.
<next slide>

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

<next slide>

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

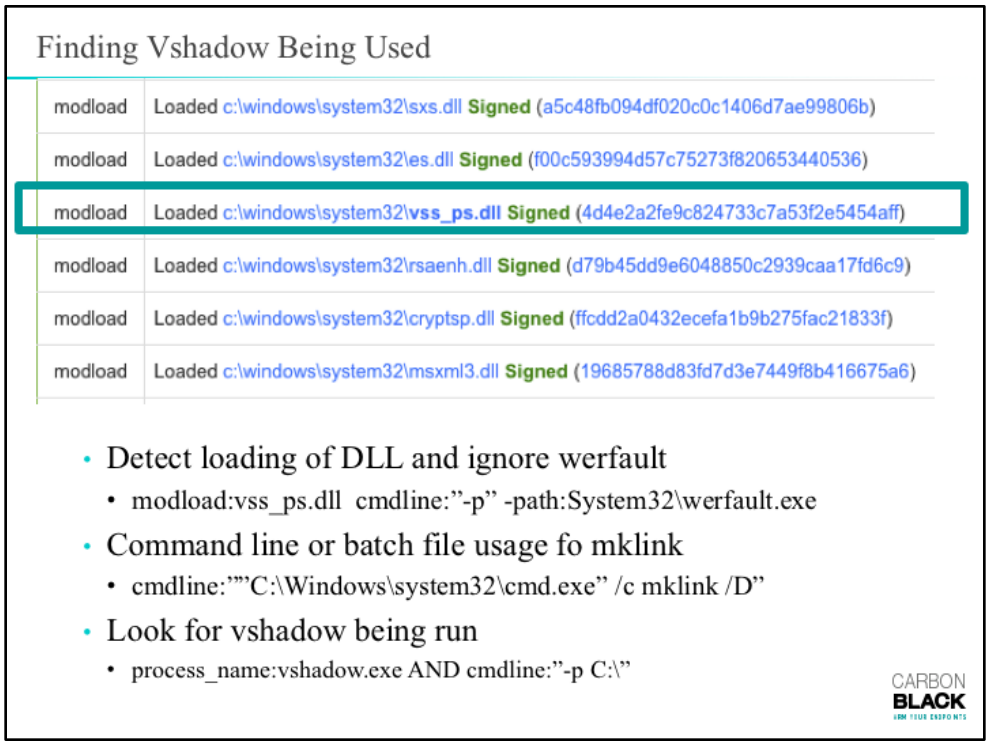
Let’s talk about that.
<next slide>



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>



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 can see is vss_ps.dll, which is a necessary component of the Volume Shadow Storage feature.

In a 3,000-host environment, a query just for vss_ps.dll 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.

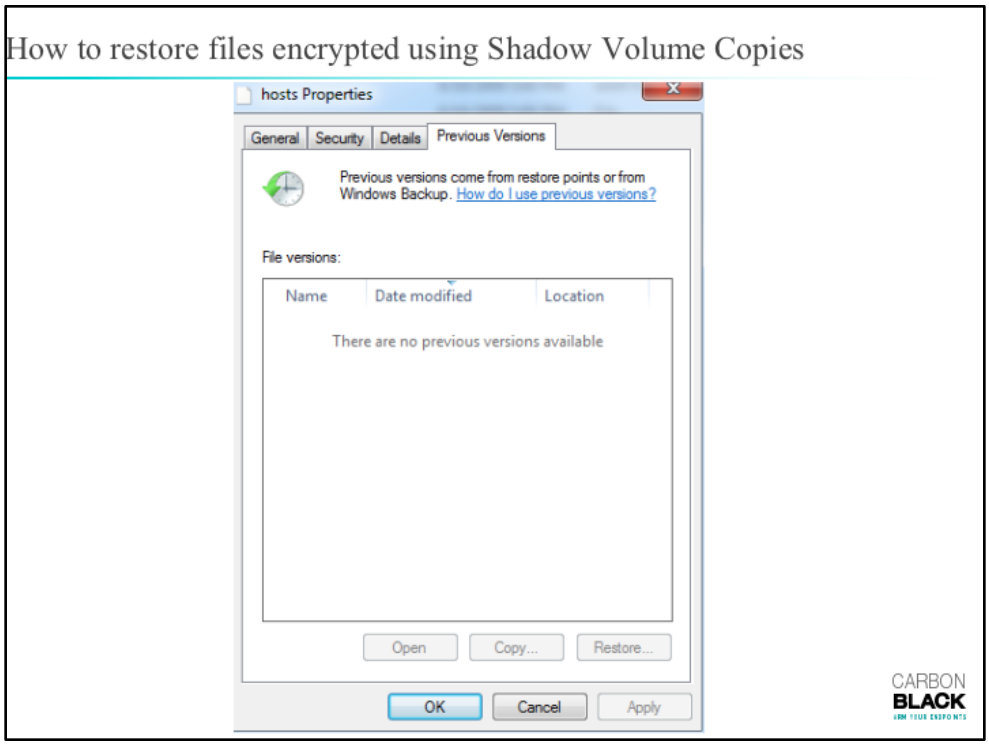
One caveat I found while researching this is the command "mklink" is a function of cmd.exe. Because of this, it is going to be hard for some IR tools to detect. luckily my IR tool sees the command line and can detect this as written in the fourth query <next slide>

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>

Let’s talk about how to defend against this type of Ransomware
<click>

<next slide>



If you have “System Restore” enabled, your system will create shadow copies that hold copies of your files from that moment back.

These copies “may,” and I use the term “may” dripping with hope, allow you to restore your files from before they were encrypted.

Using shadow copies is not foolproof and the version of the files in the shadow copy may not be the latest version and could be useless to you.

There are multiple methods online documenting how to recover with a shadow copy. The method I normally use is the native Windows option because it has the most probability of being available to me on a supported Windows system.

If possible, use whitelisting on any system you can. This will prevent the execution of any unknown binary. Similar to shot blocking
<next slide>

How to prevent this infection

CARBON BLACK

<click>
Short of white listing, you can implement rules based on your use case, unique to you environmental variables, and desired outcome.

These rules will be shown in their high-level form because even if you aren't using fancy IR tools, you can still apply this logic and most of these rules to your environment by leveraging a mix of GPO, witchcraft, and cursing.
<next slide>

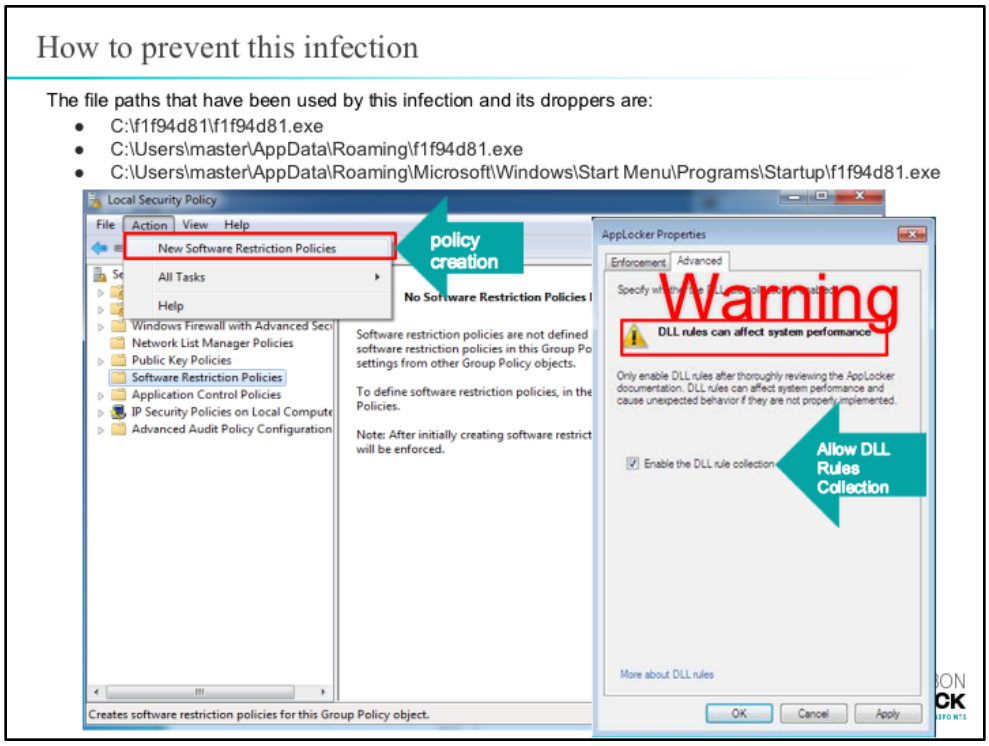
How to prevent this infection

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

CARBON BLACK

These file paths are what we are going to use as examples in our prevention steps. <next slide>



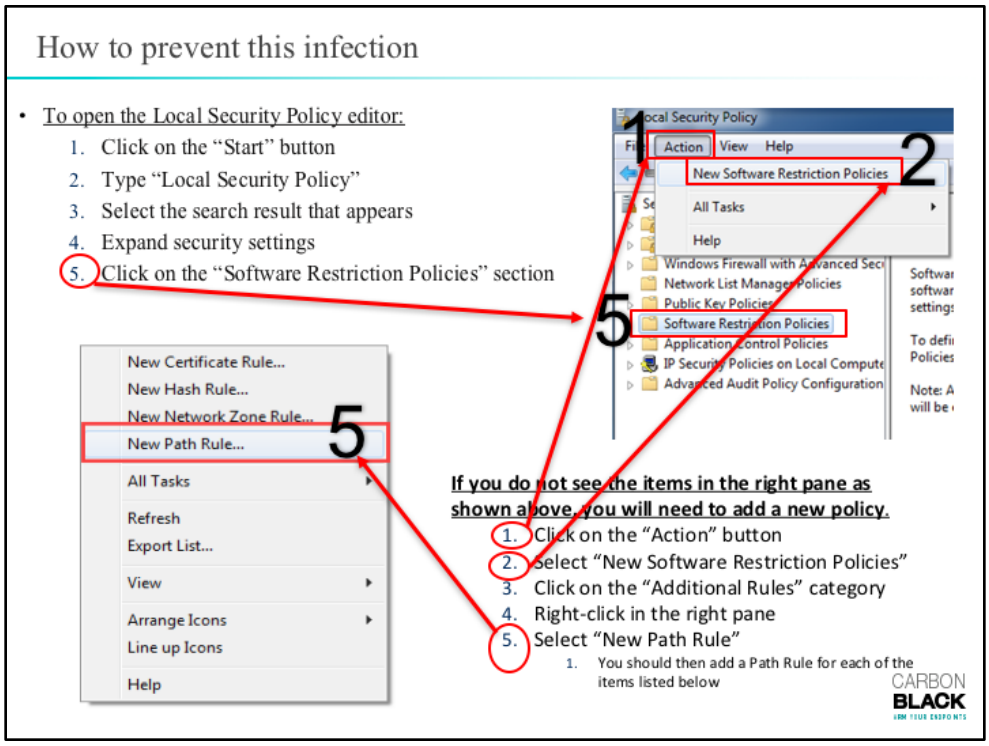
You can create a software restriction policy for a single computer using the Local Security Policy Editor or for an entire domain use the Group Policy Editor

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.

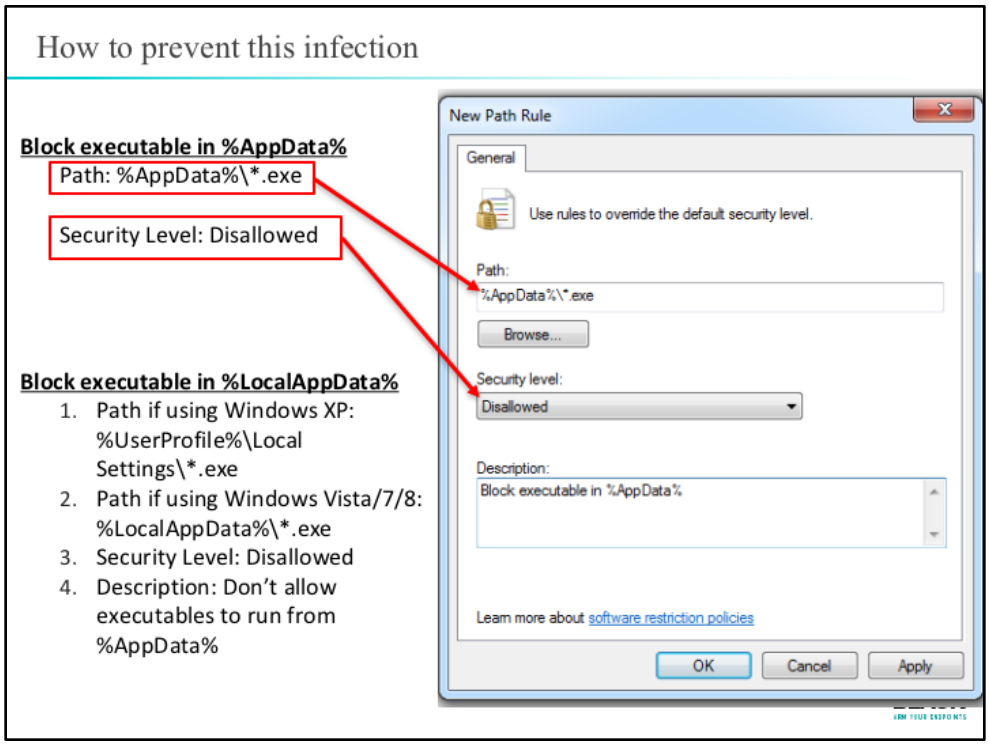
The DLL Rules Collection

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 be 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>



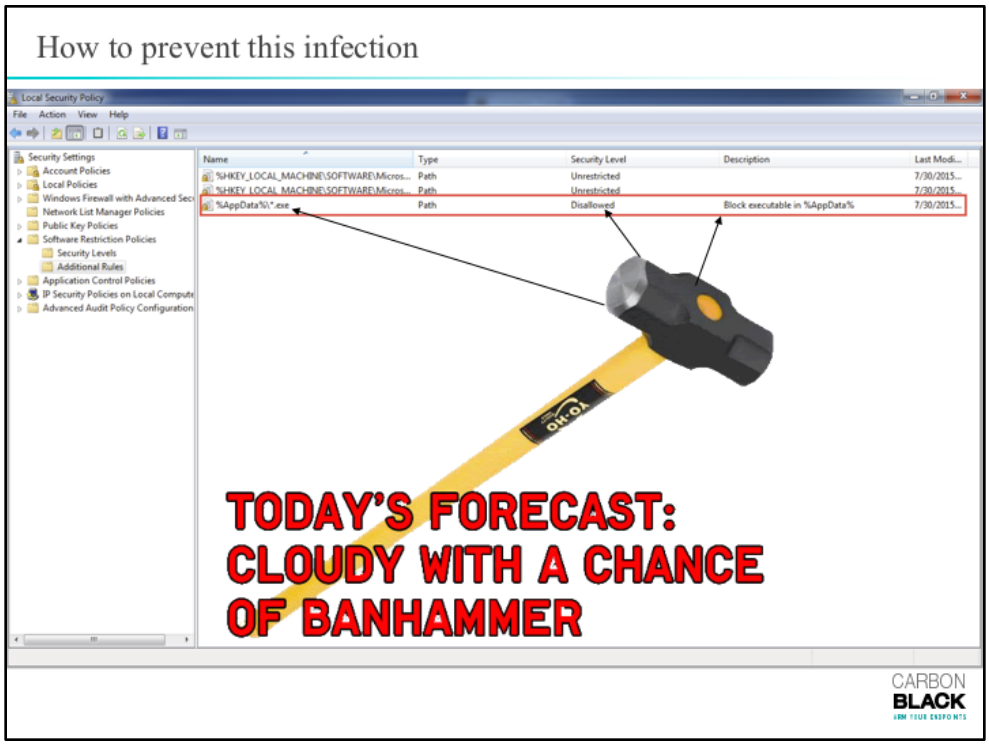
First you need to open the Security Policy editor located here and then it's just 3 quick steps to create a policy to protect your enterprise.
<click through 5>
<next slide>



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>



The final product.

<next slide>

In closing, Ransomware is annoyingly effective.

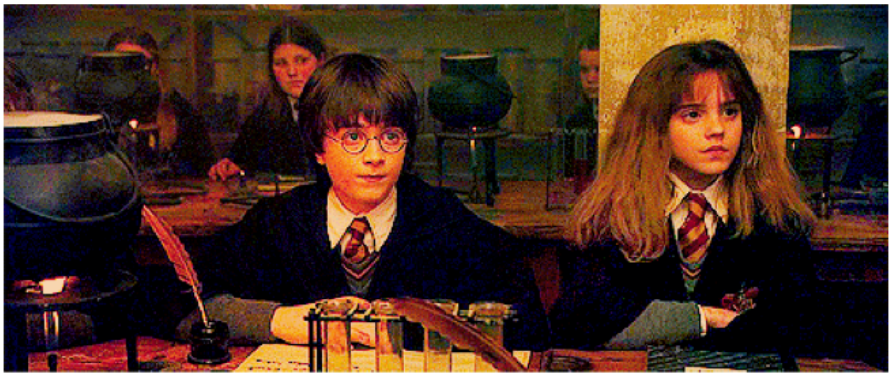
The recent additions of features such as removing shadow copies makes it even more dangerous.

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 fall for your time today and until next time, remember my motto: "Flag it, Tag it and Bag it."
<next slide>

Questions



CARBON BLACK

Questions?
<hold until questions are over>