

Remote Collections with KAPE

Triage style collection of remote system data is a fact of life these days and KAPE a excellent tool to preform this task . Remote collections can have different meaning depending on the context. Remote could mean a system on the corporate LAN or it could mean as remote system that is only available over the Internet. We will start with some basics and then show how we can apply the concepts of a LAN collection to the increasingly common situation of collection over the Internet.

In this article, we are assuming that you have a basic understanding of KAPE terms like targets and modules. Although we are showing the command line versions of these commands the concepts will work equally well the the GUI version of KAPE.

UNC Paths

Stands for "Universal Naming Convention," not just the home of the North Carolina Tar Heels. UNC is a filename format that is used to specify the location of files, folders, and resources on a local-area network (LAN). The UNC address of a file may look something like this:

```
\\server-name\directory\filename
```

We can use a UNC path to create a mapped network drive so we can access the network share as a drive letter.

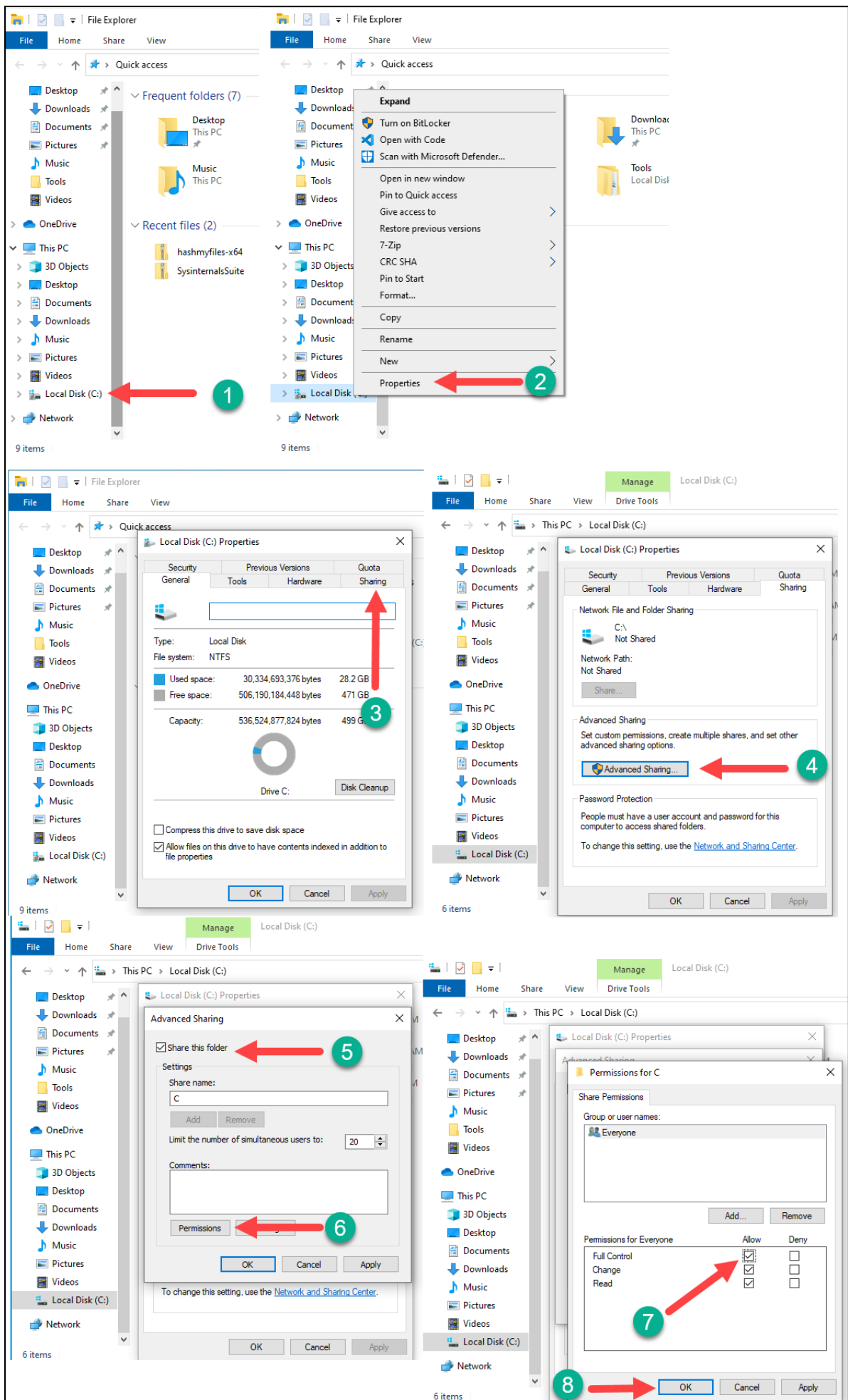
Share the C Drive on the Target

To use KAPE in this manner some setup is required. We need to be able to access the remote system's C drive. Meaning the the drive must have sharing enabled. In our examples in this paper, we are making an assumption that you have some access to the remote computer to enable the shares. In a simple Workgroup environment, that would involve these steps.

1. Select the C Drive On the Remote System
2. Right Click the C Drive and select properties
3. Click the **"Sharing"** Tab
4. Click the **"Advanced Sharing"** button
5. Check the **"Share this folder"**
6. Click the **"Permissions"** button
7. Check the **"Full Control"** box
8. Click the **"Ok"** button
9. Click the **"Ok"** button on the Advanced Sharing dialog box
10. Click the **"Close"** button

Click this [link](#) for an article on Sharing Drives in Windows.

Click this [link](#) for a graphical step by step on Sharing Drives in Windows.



Once the share has been created on the target we will create a mapped drive on the collection system to the Target share so we can access it as a drive letter.

1. Open a CMD or PowerShell session as administrator.
2. Ping the target to confirm network connectivity. Either host name or IP address is fine.

```
ping 192.168.96.162
```

or

```
ping target-1
```

3. Issue the following command to map the target's shared drive a drive letter on the collection system.

```
net use k: \\target-1\c /user:target-1\sansdfir
```

We are assigning the drive letter "k" to the UNC path \\target-1\c using the fully qualified user name on the target.

4. Issue the "net use" command to verify that the drive was mapped correctly. The "net use" command without options will show all the current mappings.

```
net use
```

5. Issue the "dir" command to verify that we can access the UNC path via the drive letter. We can use just the drive letter to access that target share.

Run KAPE using the mapped drive to the Target

When using KAPE in a remote collection, a natural first thought is how to reference that remote system in the KAPE command line. Using a UNC path or mapped drive as the --tsource parameter is how we do that. Here is an example of what that command might look like. The first example uses the UNC path and the second uses a mapped drive (that is based on a UNC path).

```
kape.exe --tsource \\target-1\c --target LnkFilesAndJumpLists --tdest c:\kape_out\test
```

```
kape.exe --tsource K:\c --target LnkFilesAndJumpLists --tdest c:\kape_out\test
```

The two commands above collect from the remote target and saves the files matching the target definition (--target) to the \kape_out\test folder on the collection system's local C drive.

Let's give this a test drive.

```
Administrator: Windows PowerShell
PS C:\trriage\kape>
PS C:\trriage\kape> .\kape.exe --tsource \\target-1\c --target LnkFilesAndJumpLists --tdest c:\trriage\kape_out\tdest --tflush
KAPE version 0.9.3-6 Author: Eric Zimmerman (kape@kroitt.com)

KAPE directory: C:\trriage\kape
Command line: --tsource \\target-1\c --target LnkFilesAndJumpLists --tdest c:\trriage\kape_out\tdest --tflush

System info: Machine name: KAPE-SERVER-VM, 64-bit: True, User: analyst OS: Windows10 (10.0.19041)

Using Target operations
  Flushing target destination directory 'c:\trriage\kape_out\tdest'
  Creating target destination directory 'c:\trriage\kape_out\tdest'
Found 6 targets. Expanding targets to file list...
VSCs are not supported over UNC paths. Disabling --vss ←
Found 24 files in 0.199 seconds. Beginning copy...

Copied 23 (Deduplicated: 1) out of 24 files in 1.0043 seconds. See '*_CopyLog.csv' in 'c:\trriage\kape_out\tdest' for copy details
Total execution time: 1.0168 seconds

PS C:\trriage\kape>
```

The Copy Log from this run of KAPE shows what files were collected. There are two other logs created, skipped log and console log. Skipped log shows you the files that met the target search criteria but were not copied for some reason and the console log show the same thing that is in the screenshot above.

Timeline Explorer v1.1.0.0				
File Tools Tabs Help				
2020-09-11T04:02:47_CopyLog.csv				
Drag a column header here to group by that column				
Line	Tag	Copied Timestamp	Source File	File Size
1		2020-09-11 04:02:49.0360521	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\desktop.ini	432
2		2020-09-11 04:02:49.0865680	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\Downloads.Ink	451
3		2020-09-11 04:02:49.1036382	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\hashmyfiles-x64.Ink	627
4		2020-09-11 04:02:49.1246284	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\kape.Ink	612
5		2020-09-11 04:02:49.1445444	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\kape_out.Ink	729
6		2020-09-11 04:02:49.1624621	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\Local Disk (C:) (2).Ink	386
7		2020-09-11 04:02:49.1917700	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\ms-gamingoverlay--kgi-check-.Ink	172
8		2020-09-11 04:02:49.2084973	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\SysInternalSuite.Ink	637
9		2020-09-11 04:02:49.2256214	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\The Internet.Ink	104
10		2020-09-11 04:02:49.2396820	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\This PC.Ink	104
11		2020-09-11 04:02:49.2557005	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\Tools.Ink	519
12		2020-09-11 04:02:49.2736002	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\5f7b5f1e01b83767...	4608
13		2020-09-11 04:02:49.2914633	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\dd7c3b1ad01c168b...	3072
14		2020-09-11 04:02:49.4054984	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\fb1b4d95cf55d32a...	14336
15		2020-09-11 04:02:49.4266342	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\1ced32d74a95c7bc...	2008
16		2020-09-11 04:02:49.4435637	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\590aee7bdd69b59b...	5444
17		2020-09-11 04:02:49.4599546	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\7e4dca80246863e3...	24
18		2020-09-11 04:02:49.4786145	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\9d1f905c5844aee...	1780
19		2020-09-11 04:02:49.4964528	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\c81827d56ff89856...	4636
20		2020-09-11 04:02:49.5314841	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\ccba5a5986c77e43...	22236
21		2020-09-11 04:02:49.5465143	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\fb1b4d95cf55d32a...	24
22		2020-09-11 04:02:49.5625199	\\target-1\c\users\sansdffr\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\fb18460fde0109990...	24
23		2020-09-11 04:02:49.5835560	\\target-1\c\users\sansdffr\Desktop\Visual Studio Code.Ink	1413

In the Copy Log we see Ink files like we would expect from the command line we just executed. We got what we were expecting. Important point of interest, volume shadow copies can't be copied over a UNC path.

Now lets try a different target, lets grab some registry files. We are selecting these files to demonstrate another important limitation of using a UNC path to the target system in addition to the VSC limitation. The Registry files are protected files locked by Windows.

```
kape.exe --tsource \\remote-target-name\c --target RegistryHives --tdest
c:\kape_out\test
```

```
Administrator: Windows PowerShell
PS C:\trriage\kape>
PS C:\trriage\kape> .\kape.exe --tsource \\target-1\c --target RegistryHives --tdest c:\trriage\kape_out\tdest --tflush
KAPE version 0.9.3.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: C:\trriage\kape
Command line: --tsource \\target-1\c --target RegistryHives --tdest c:\trriage\kape_out\tdest --tflush

System info: Machine name: KAPE-SERVER-VM, 64-bit: True, User: analyst OS: Windows10 (10.0.19041)

Using Target operations
  Flushing target destination directory 'c:\trriage\kape_out\tdest'
  Creating target destination directory 'c:\trriage\kape_out\tdest'
Found 2 targets. Expanding targets to file list...
VSCs are not supported over UNC paths. Disabling --vss
Found 9 files in 0.421 seconds. Beginning copy...
  Deferring '\\target-1\c\users\sansdfir\NTUSER.DAT' due to IOException...
  Deferring '\\target-1\c\users\sansdfir\ntuser.dat.LOG1' due to IOException...
  Deferring '\\target-1\c\users\sansdfir\ntuser.dat.LOG2' due to IOException...
  Deferring '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat' due to IOException...
  Deferring '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1' due to IOException...
  Deferring '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2' due to IOException...
Deferred file count: 6. Copying locked files...
Could not copy file '\\target-1\c\users\sansdfir\NTUSER.DAT' to 'c:\trriage\kape_out\tdest\target-1\c\users\sansdfir\NTUSER.DAT'. Error: R
Could not copy file '\\target-1\c\users\sansdfir\ntuser.dat.LOG1' to 'c:\trriage\kape_out\tdest\target-1\c\users\sansdfir\ntuser.dat.LOG1'
Could not copy file '\\target-1\c\users\sansdfir\ntuser.dat.LOG2' to 'c:\trriage\kape_out\tdest\target-1\c\users\sansdfir\ntuser.dat.LOG2'
Could not copy file '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat' to 'c:\trriage\kape_out\tdest\target-1\c\us
opies across UNC paths are not supported
Could not copy file '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1' to 'c:\trriage\kape_out\tdest\target-1
ror: Raw copies across UNC paths are not supported
Could not copy file '\\target-1\c\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2' to 'c:\trriage\kape_out\tdest\target-1
ror: Raw copies across UNC paths are not supported

WARNING: THERE WERE 6 FILE COPY FAILURES! See the console log for details!!!

Copied 3 out of 9 files in 0.9273 seconds. File copy errors: 6. See console log for details! See '*_CopyLog.csv' in 'c:\trriage\kape_out\t
Total execution time: 0.9406 seconds
PS C:\trriage\kape>
```

In this screenshot we see two things. The first is that KAPE is deferring the copy of certain files, these are the protected files. The protected files will be copied later in the process using a different method called raw copy. The raw copy gets around the system locks. Unfortunately Windows does not allow this type of copy across a UNC path. Bummer. It is important to point out that this is not a KAPE error or limitation of KAPE, this is a file system limitation. Whatever the reason, it keeps us from collecting what we need using this approach. Lets look at a different approach, still using UNC paths, that will work on these protected files, all file types in fact.

Run KAPE from the Target

Lets looks at this problem from a different angle. The issue is that attempting to copy the protected files from the target across the UNC path with a raw copy. What if we process the files on the target and copy the resulting, unprotected files, across the UNC path? They are unprotected because they are just copies of the original files. We will also execute the kape.exe program from its location on the collection system so that we don't have to copy any files to the target system. This minimizes the footprint we leave behind on the target. There will still be footprints (artifacts) because of the program execution. There will be a Prefetch file created or updated and other Window Registry entries. But, we won't overwrite anything that might be in unallocated space. More on that in a minute.

Where is how this works.

1. From the collection system, share the folder on the collection system that contains the kape.exe executable. We are going to map that share to a drive letter on the target. Basically, we are doing the opposite of what we did earlier to share the Target's hard drive. We are also going to save the results of our collection to subfolder under the kape folder on the collection system because we are lazy. This way we don't need to create two shares.
2. On the target system use the **"net use"** command as we did above to map the kape folder on the collection system to a drive letter on the Target system.

```
net use k: \\kape-server-vm\trriage /user:kape-server-vm\analyst
```

3. Verify the share and the mapped drive on the target as we did for the server above.

```
Windows PowerShell
PS C:\Users\sansdfir> net use k: \\kape-server-vm\trriage /user:kape-server-vm\analyst
The command completed successfully.

PS C:\Users\sansdfir> net use
New connections will be remembered.

Status      Local      Remote      Network
-----
OK          K:         \\kape-server-vm\trriage  Microsoft Windows Network
The command completed successfully.

PS C:\Users\sansdfir> dir k:

Directory: k:\

Mode                LastWriteTime         Length Name
----                -
d-----          9/10/2020  11:02 PM             kape
d-----          9/10/2020  11:21 PM          kape_out
d-----          8/12/2020   4:15 PM          kape_sftp

PS C:\Users\sansdfir> |
```

Now we are going to run a very similar kape command to collect the protected Registry Files from the Target and save the collected files to the collection system. The approach does not write any collected data or temp files to the target. This is because the actual kape.exe is kept on the collection system and the collected data is written directly over the UNC path to the collection system without ever being written to the target machine. Oh, and we can grab the Volume Shadow Copies too.

```
k:\kape\kape.exe --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss
```

```
Administrator: Windows PowerShell
PS C:\Users\sansdfir> k:\kape\kape.exe --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss
KAPE version 0.9.3.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: k:\kape
Command line: --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss

System info: Machine name: TARGET-1, 64-bit: True, User: sansdfir OS: Windows10 (10.0.19041)

Using Target operations
  Creating target destination directory 'k:\kape_out\tdest'
Found 2 targets. Expanding targets to file list...
VSSs found on volume C: 3. Mounting...
VSS 1   (Id {6c5b95c0-e590-4f16-894c-cb0f515db7fd}, Created on: 2020-08-25 16:30:31.6034630 UTC) mounted OK!
VSS 2   (Id {5e5c58f7-c21d-4922-890a-76cad6542720}, Created on: 2020-09-07 21:38:34.5482850 UTC) mounted OK!
VSS 3   (Id {d8953257-be6d-419b-af9c-1db54eeae225}, Created on: 2020-09-09 23:34:49.8928090 UTC) mounted OK!
Found 120 files in 0.795 seconds. beginning copy...
Deferring 'C:\Windows\System32\config\SAM.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SAM.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\SECURITY.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SECURITY.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\SOFTWARE.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SOFTWARE.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\SYSTEM.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SYSTEM.LOG2' due to IOException...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat' to 'k:\kape_out\tdest\C\us
ers\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat'. Hashing source file...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1' to 'k:\kape_out\tdest
\C\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1'. Hashing source file...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2' to 'k:\kape_out\tdest
\C\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2'. Hashing source file...

Copied 108 (Deduplicated: 12) out of 120 files in 20.9991 seconds. See '*_CopyLog.csv' in 'k:\kape_out\tdest' for copy d
etails

Total execution time: 21.0117 seconds

PS C:\Users\sansdfir> |
```

Be careful - KAPE will write files to the Target

In certain situations, when running KAPE from a target system, KAPE will write its collection files to the Target system even if the ultimate destination is somewhere off the target system. These files may be just temporary and will be deleted after the files have been sent to another destination. KAPE has some great features that allow target files to be written to an SFTP server, Amazon S3, and Azure. KAPE can even be the SFTP server itself. For these features to work, these files must first be written somewhere and the location is often the Target system itself.

The easiest way to see if this situation is going to occur is to look at the command line you are going to run. If the **--tdest** option or the **--mdest** option is a location on the Target's OS drive you are going to have the collected files first written to the target. Let's look at a command line run from the target for sending the collected files to an SFTP server.

```
kape.exe --tsource C: --target !SANS_Triage --tdest C:\temp\tout --scs  
104.248.94.196 --scp 22 --scu kape-ssh --scpw "KAP3g0at" --vhdx %m_kape_collect
```

What's happening in the command line above? First thing is that we have used a **--tdest** option that is a location on the Target system. In this case we have no other choice. Additionally, the creation of the VHDX, which is required when we use the **--scs** option, and the subsequent zipping up of the VHDX will write to the Target system. We are doing a triage collection, files collected are defined by the !SANS_Triage target, the files are going to be sent to an SFTP server at 104.248.94.196. The files will be packaged up in a VHDX and further zipped up to reduce the transfer time. This is a very useful option of KAPE as it allows us to get the collected files out to a secure server where the analyst team can get to work on them right away.

I don't want to make too big of a deal out of this but there is a risk in running the command above. The files will first be written to the Target system. This may not matter or it may be a trade off. The trade off being that we are betting that getting the files to a secure location is far more beneficial than any risk of potentially losing artifacts in unallocated space. There may be no way of getting a full forensic image of this system so we don't worry about it. The point is that you are knowingly making this decision.

But, and I have been in this situation, we don't have any reason to think that unallocated space is a factor until after we analyze the triage data and we do in fact have the option of doing a full image. If this is the case what can we do to avoid the potential of overwriting evidence in unallocated? We will cover one solution in the next section.

Remote KAPE Collections across the Internet

The techniques of running the kape.exe executable from a network share and writing the collected files to a network share are simple but very powerful. We have shown this technique using very narrowly focused targets but it could be an entire triage collection. It was simple to create the shares because in our demo we were in a LAN situation. We had various ways to create the shares through something like PowerShell or even logging into the target system via RDP. How do we do this over Internet, over a WAN?

The technique I'm going to describe is to use a free, open source service called ZeroTier One. Basically ZeroTier allows us to create a Software Defined Wide Area Network (SD-WAN) in minutes. The ZeroTier network that we are going to define will work exactly like the examples that we have shown above.

Yes, we are going to install a very small application on the target but we are not going to write KAPE or any of the collected files to the target. This is a key point. There have been some very clever techniques published for getting the KAPE executable to the target so that an automated, scripted collection can be easily run. **Check out these articles for info on those methods.** Those techniques work exactly as advertised but... they write all the collected data to the target system before forwarding them on to their ultimate destination on an sftp server, Amazon S3, Microsoft Azure. As discussed earlier, if there is an intention to do a full forensic image of the target at some point or even the possibility of needing the full image, we really don't want to overwrite unallocated unless there is no other choice. So, how does the work.

All we need to do is:

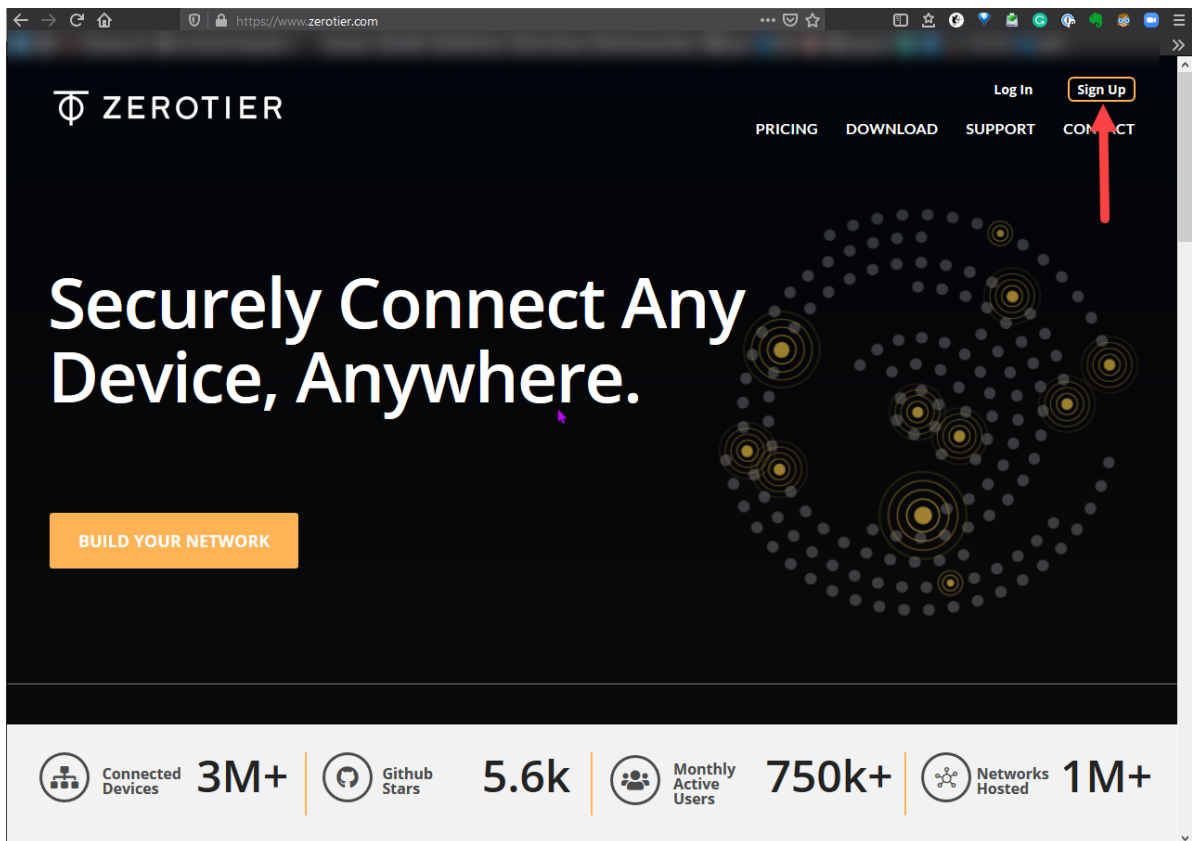
1. Create a ZeroTier Account. This account can be used for many collections and we can create up to 100 networks with this free account. We will remove the network after the collection.
2. Install ZeroTier on both the collection system and the remote system.
3. Join both the collection system and the remote system to the ZeroTier network we setup for this collection.
4. Approve the nodes for both systems once they join the network. This is done from a web console on the collection system.
5. Map the drive from the collection system, our destination for the collected files, exactly as we did in the LAN environment.
6. Run the KAPE collection from the Target.

Getting ZeroTier installed on the system is going to require some assistance to get control of the Target system. This is going to be the case with most remote collections so this is not unique to using ZeroTier. If already installed, we could use Zoom or some other software that will allow us to remotely control the system. Possibly a remote IT person or even the custodian (the person to whom the system belongs) themselves can help with the ZeroTier installation if screen sharing is not available. Once the target is connected to the ZeroTier system, we can use RDP to access the the remote Target over the ZeroTier network. Regardless of the how we accomplish it, we need to get ZeroTier installed and connected to the ZeroTier network. This is a very simple procedure.

Let's walk through this step by step.

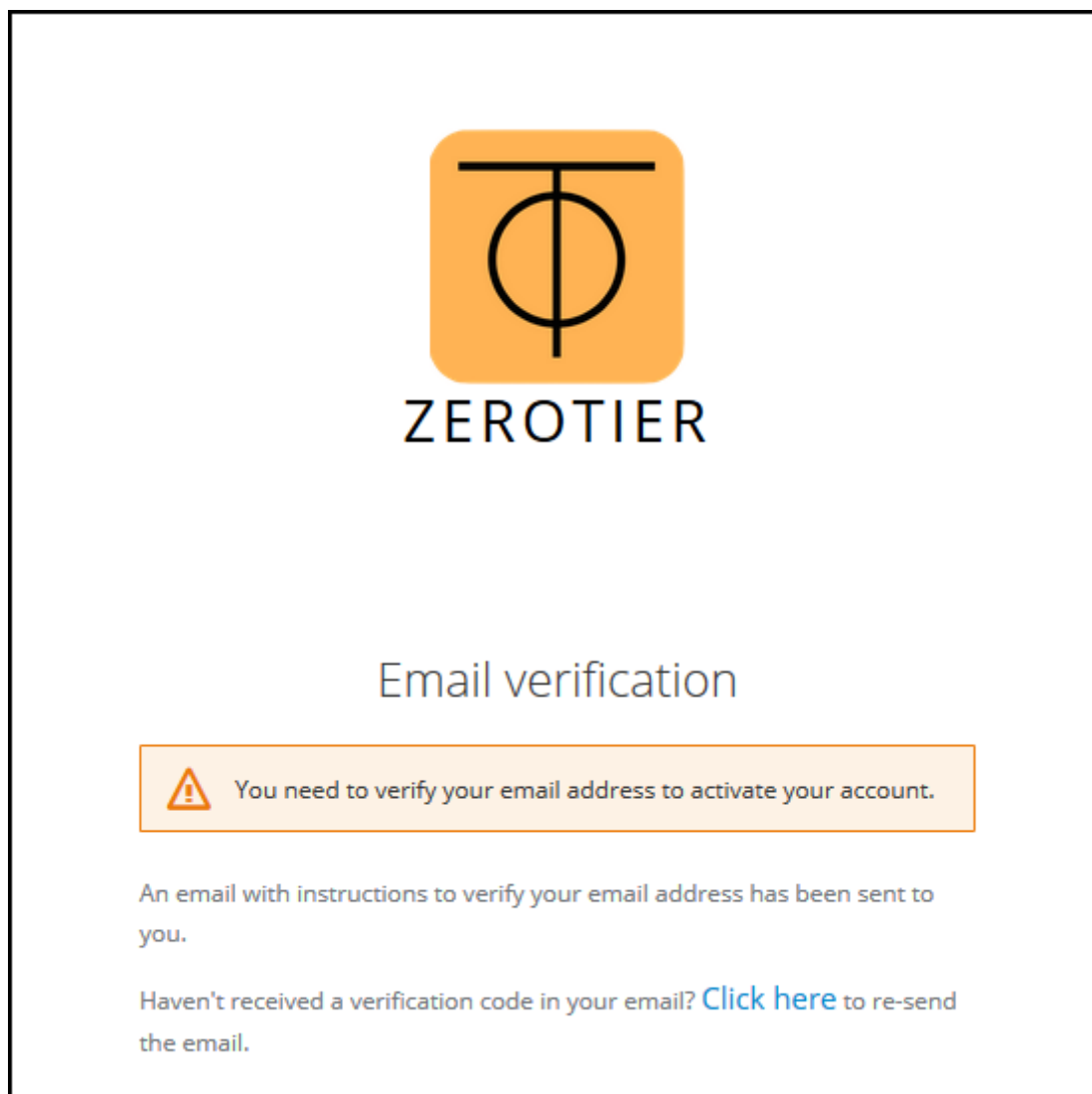
Create a ZeroTier Account

In your web browser navigate to zerotier.com and select "Sign Up".

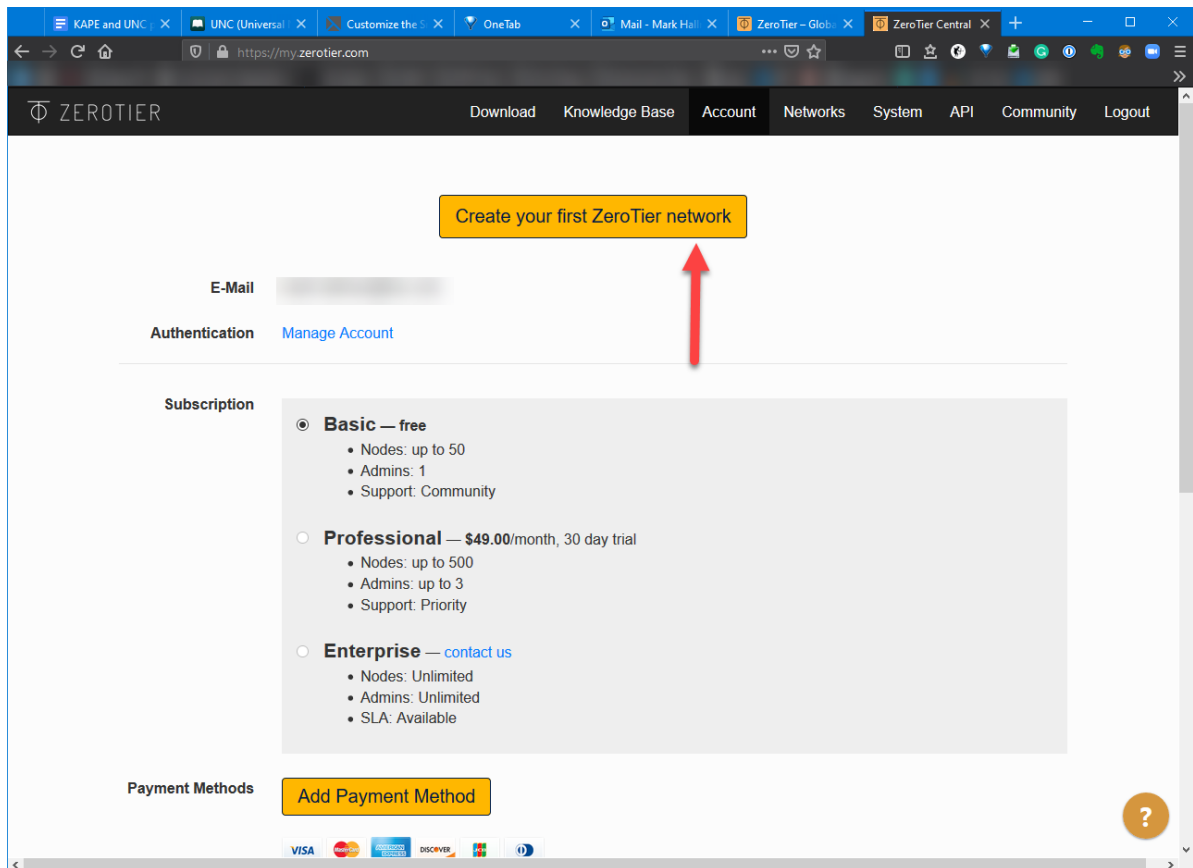


Complete the registration page and click "register".

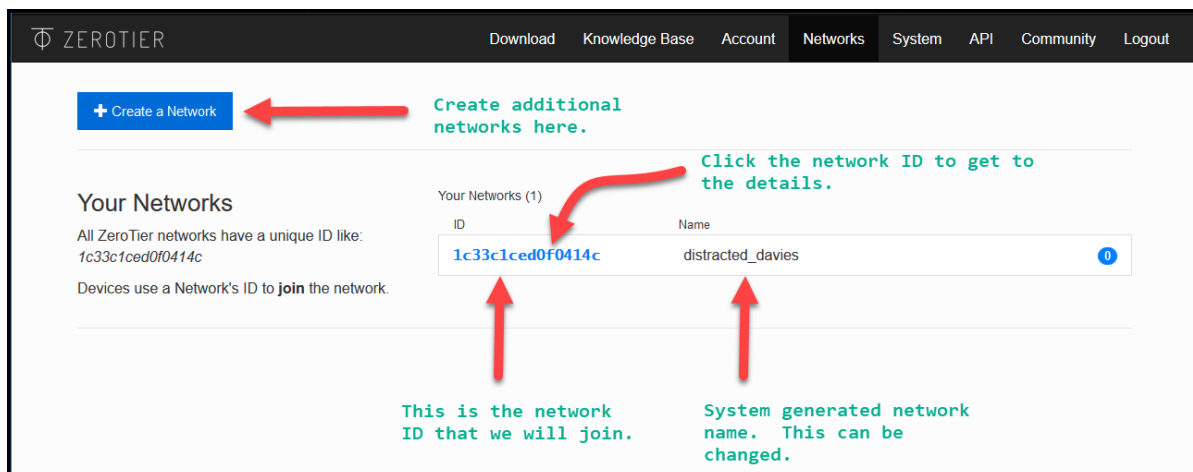
Check your email and activate your new ZeroTier account.



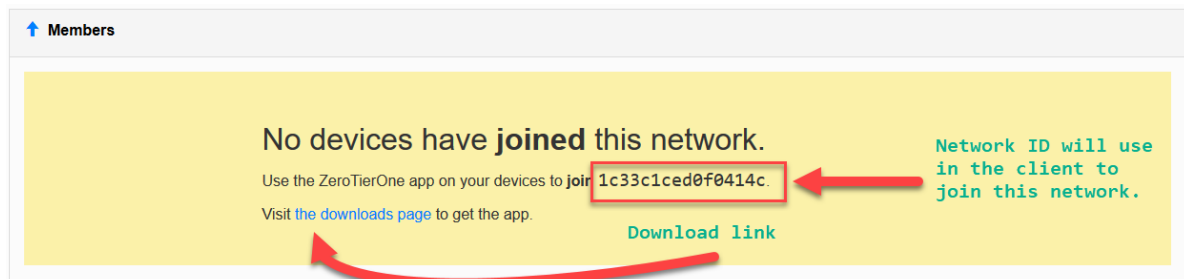
Log into ZeroTier for the first time and click "Create your first ZeroTier Network"



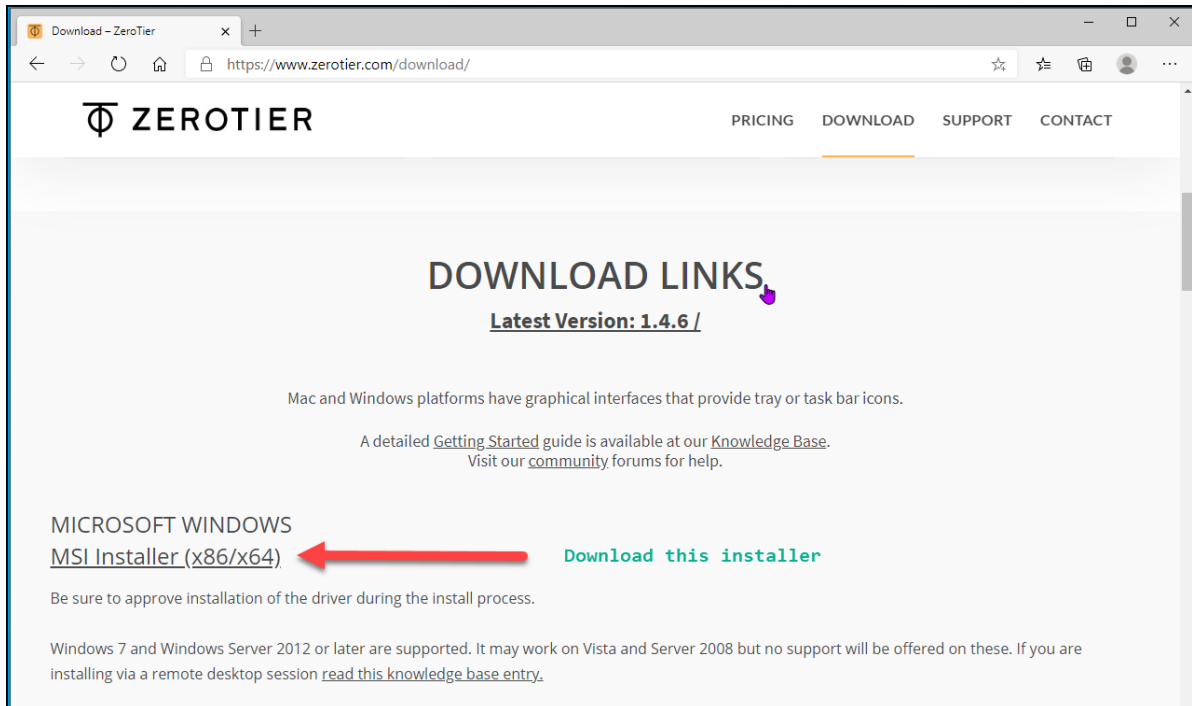
You will now see your first network. Click the Network ID link to get to the network details page. You will not need to make any changes on this page at this time but please explore if you wish to see the configuration options.

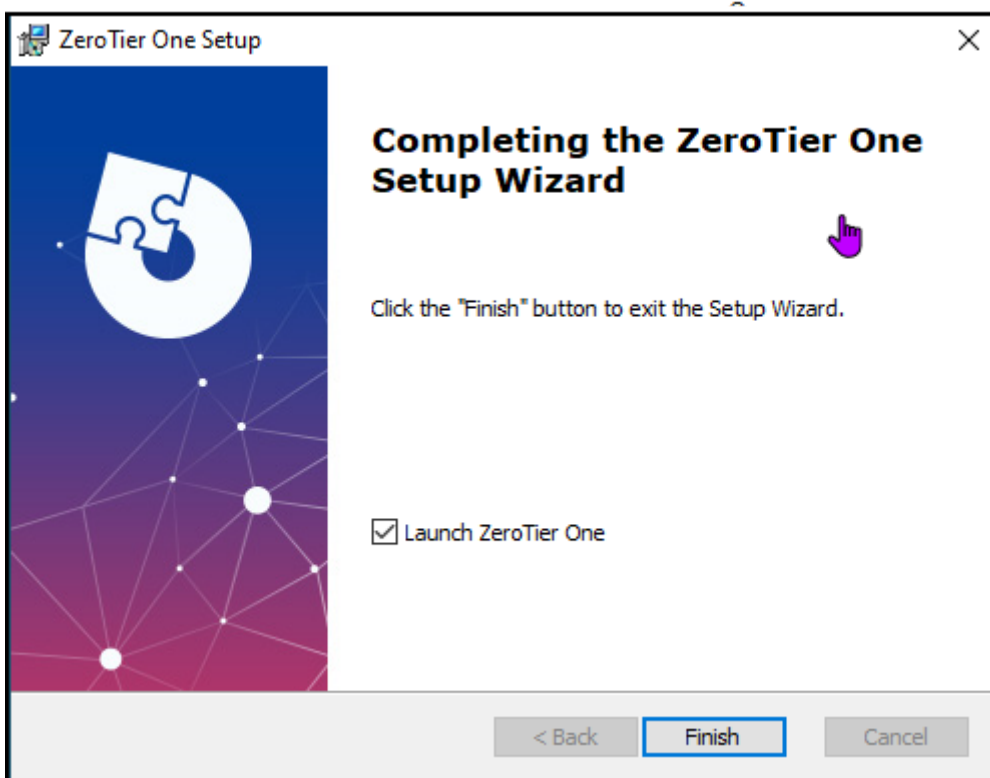
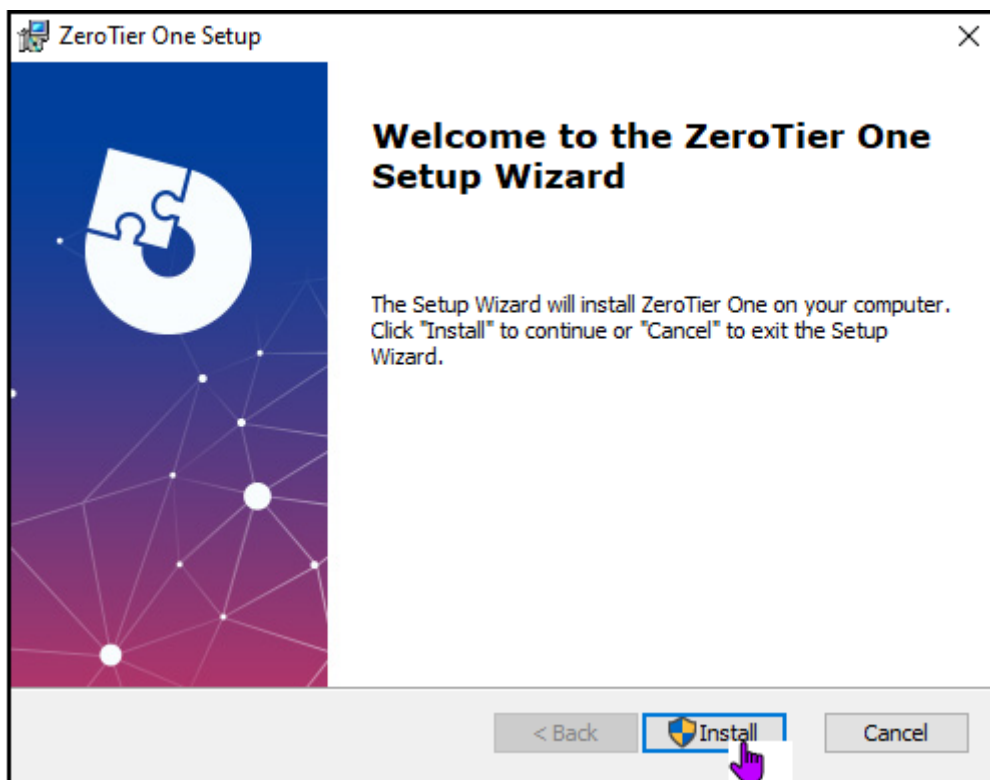


Next we need to download and install the ZeroTier app on both the collection system and the Target. The download links can be found in Several locations on the ZeroTier site. Since we are on the network details page, we can get the download link here so we can download the ZeroTier client and install in both the collection system and the Target.

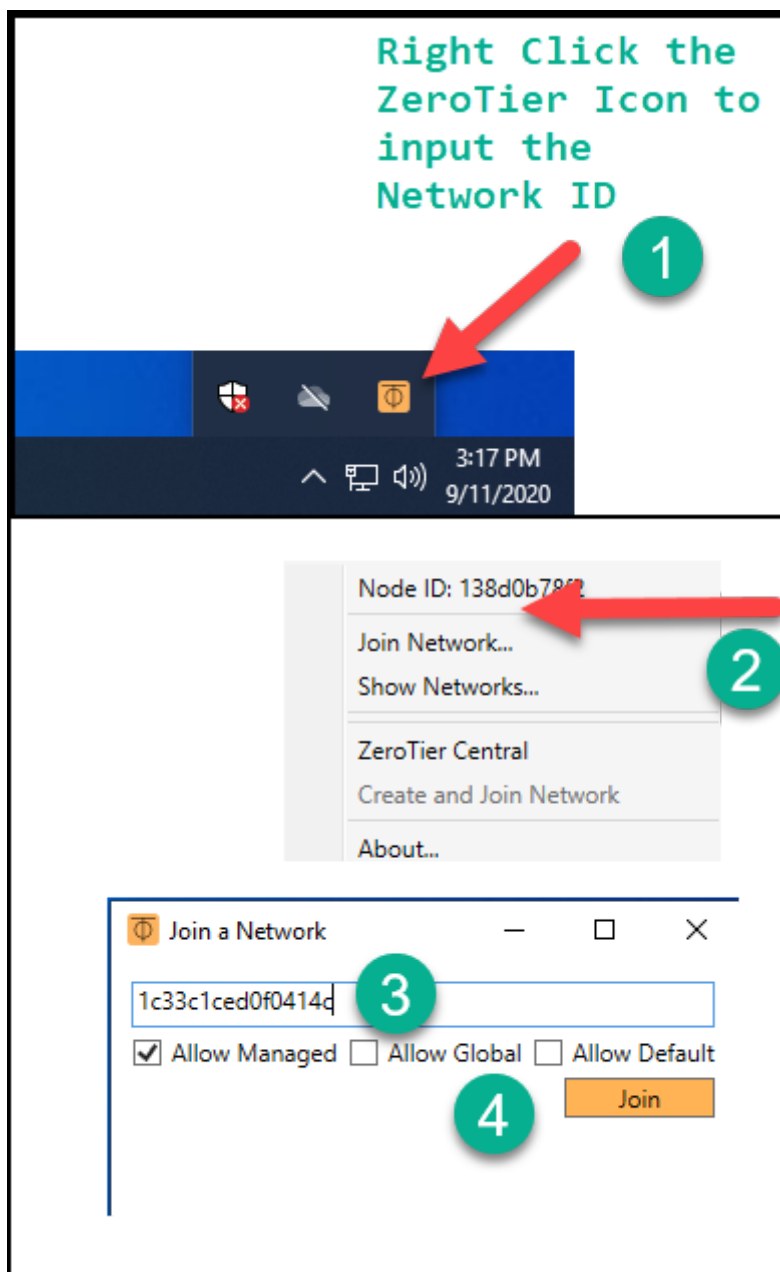


From the ZeroTier downloads page, download the Windows installer. Once downloaded, double click the installer in your Downloads folder and run the installer accepting all the defaults.





Click "Finish" and the ZeroTier client will be launched and minimized to your Windows System Tray. Right click the ZeroTier Icon in the system tray, select "Join Network", enter the Network ID, and click the "Join" button. The system will now connect to the Network. The system will have to be authorized from the ZeroTier system console (where we got the network ID). Remember, perform this download, install and configuration for both the collection system and the target. The collection system could/should be setup ahead of time. Make note of the Node ID so you can label the collection system and the target in the management console.



When the system joins the network you will see this Windows notification, select "Yes".

Networks



Network 5

Do you want to allow your PC to be discoverable by other PCs and devices on this network?

We recommend allowing this on your home and work networks, but not public ones.

Yes

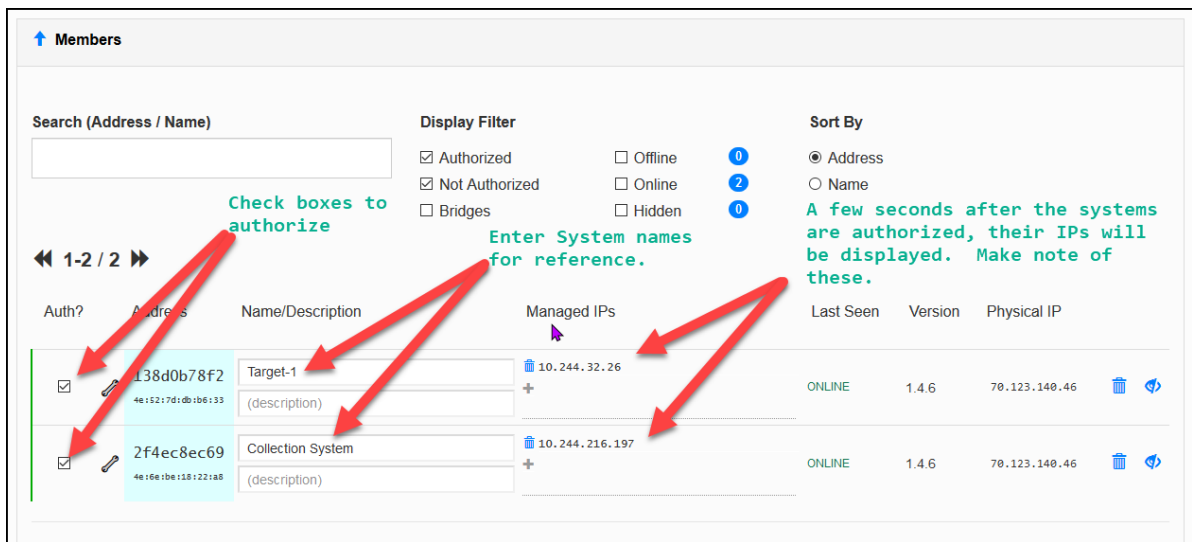
No

Now we need to authorize the system on the network via the ZeroTier web-based management console. Login to your ZeroTier account and select **"My Networks"**. Then select the network ID that we have joined the two systems to. Scroll down to the **"Member"** section. In our example, only the two systems will be displayed. Your configuration could have more if you have added more systems to this network.

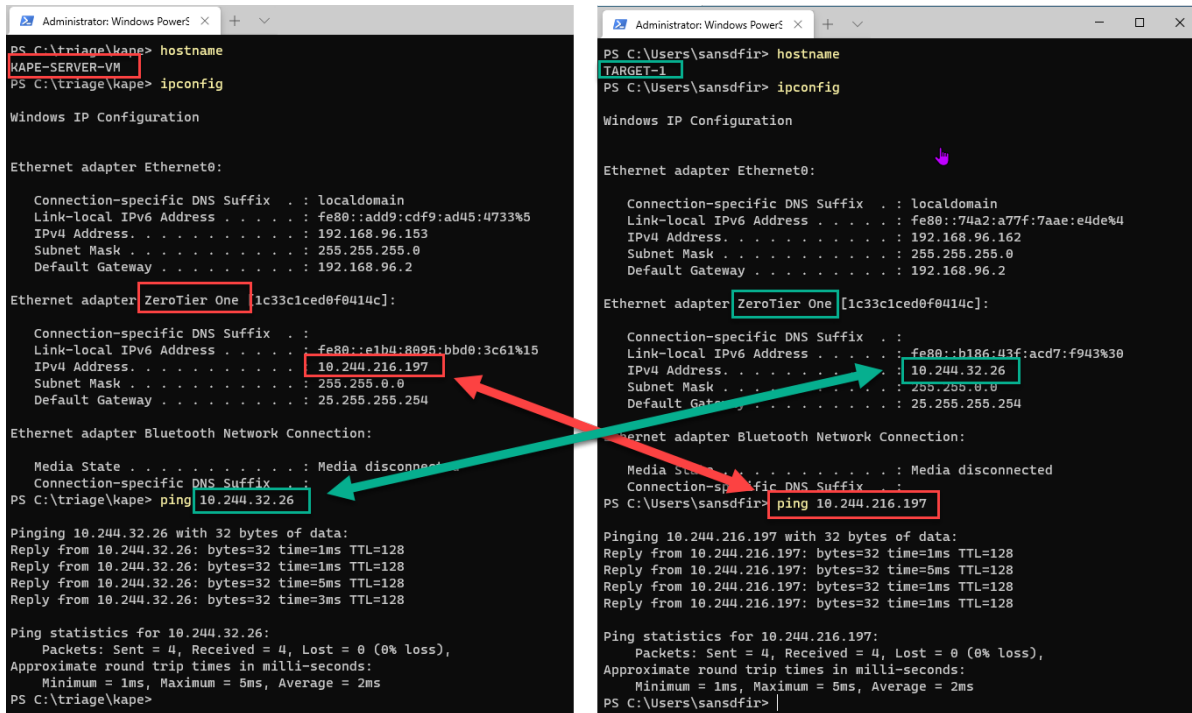
Check the boxes to the left of each of the systems to authorize them.

Add a description to each system so you know which is which based upon the Node ID you noted when you configured the client.

Make note of the IPs assigned to each system. You will need these to map the shares and RDP to the systems.



At this point both systems are live on the private network. You can verify this by pinging from one to the other, both directions.



Now you need to duplicate the steps that we did above when we shared the the collection system folder containing the kape.exe executable and then on the target system map that share to a drive letter.


```
Administrator: Windows PowerShell
PS C:\Users\sansdfir> hostname
TARGET-1
PS C:\Users\sansdfir> net use k: \\10.244.216.197\triage /user:kape-server-vm\analyst
The command completed successfully.

PS C:\Users\sansdfir> net use
New connections will be remembered.

Status      Local      Remote      Network
-----
OK          K:         \\10.244.216.197\triage  Microsoft Windows Network
The command completed successfully.

PS C:\Users\sansdfir> dir k:

Directory: K:\

Mode                LastWriteTime         Length Name
----                -
d-----          9/10/2020   11:02 PM             kape
d-----          9/11/2020   12:00 AM          kape_out
d-----          8/12/2020    4:15 PM          kape_sftp

PS C:\Users\sansdfir> |
```

Now we can run the KAPE collection from the Target, executing the kape.exe from the collection system and writing the files directly to the collection system. This is the exact same command that we used above (because we mapped to the same drive letter).

```
k:\kape\kape.exe --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss
```

```
Administrator: Windows PowerShell
PS C:\Users\sansdfir>
PS C:\Users\sansdfir> k:\kape\kape.exe --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss
KAPE version 0.9.3.0 Author: Eric Zimmerman (kape@kroll.com)

KAPE directory: k:\kape
Command line: --tsource C --target RegistryHives --tdest k:\kape_out\tdest --vss

System info: Machine name: TARGET-1, 64-bit: True, User: sansdfir OS: Windows10 (10.0.19041)

Using Target operations
  Creating target destination directory 'k:\kape_out\tdest'
Found 2 targets. Expanding targets to file list...
VSCs found on volume C: 3. Mounting...
VSS 1 (Id {6c5b95c0-e590-4f16-894c-cb0f515db7fd}, Created on: 2020-08-25 16:30:31.6034630 UTC) mounted OK!
VSS 2 (Id {5e5c58f7-c21d-4922-890a-76cad6542720}, Created on: 2020-09-07 21:38:34.5482850 UTC) mounted OK!
VSS 3 (Id {d8953257-be6d-419b-af9c-1db54eeae225}, Created on: 2020-09-09 23:34:49.8928090 UTC) mounted OK!
Found 120 files in 0.795 seconds. beginning copy...
Deferring 'C:\Windows\System32\config\SAM.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SAM.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\SECURITY.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SECURITY.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\SOFTWARE.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\SOFTWARE.LOG2' due to IOException...
Deferring 'C:\Windows\System32\config\DEFAULT.LOG1' due to IOException...
Deferring 'C:\Windows\System32\config\DEFAULT.LOG2' due to IOException...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat' to 'k:\kape_out\tdest\C\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat'. Hashing source file...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1' to 'k:\kape_out\tdest\C\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG1'. Hashing source file...
Copied deferred file 'C:\Users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2' to 'k:\kape_out\tdest\C\users\sansdfir\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2'. Hashing source file...

Copied 108 (Deduplicated: 12) out of 120 files in 20.9991 seconds. See '*_CopyLog.csv' in 'k:\kape_out\tdest' for copy details

Total execution time: 21.0117 seconds

PS C:\Users\sansdfir> |
```

There you have it, remote collections with KAPE over a private network that you created in minutes. Create an ZeroTier account and a couple of VMs and test this out. Enjoy!

