AMP for Endpoints SF-EICAR

Last Updated: August 9, 2016

CHAPTER 1
INTRODUCTION

This attack scenario mirrors many of the targeted attacks seen in the wild to install an advanced persistent threat (APT) on a target computer. Frequently these attacks are thought of as targeting high level executives, but any user with access to confidential information is fair game - QA frequently has access to intellectual property like source code, accountants have access to financial records, HR has access to employee records, and sales has access to customer records.

Attackers often begin reconnaissance of the target organization by combing through public records for information. This can include corporate websites, Internet forums, social networking sites, and in the case of publicly traded companies various shareholder information. In our scenario we will see how AMP for Endpoints can allow an administrator to discover not only the initial targeted attack, but also the subsequent activity.

IMPORTANT! In the following scenario the policy for the AMP for Endpoints Connector was set to audit-only mode to show the full range of actions malicious files could take and how each action is recorded and displayed by AMP for Endpoints.

CHAPTER 2
THE ATTACK

The attack starts when the victim opens a PDF file that exploits CVE-2010-2883, the Adobe SING Table Parsing Stack Buffer Overflow, a 0-day vulnerability that was exploited in the wild in September 2010. Like many attacks, the exploit sets off a chain of events including downloading and executing fake malware, which will show up in the AMP for Endpoints console and can be used to demonstrate how AMP for Endpoints can be used to detect and remediate such attacks.

In a typical scenario the attacker would probably send the victim an email message with the malicious PDF file attached or with an embedded link pointing to the PDF hosted elsewhere. This is fairly typical of a spear phishing attack.

When it is opened, the PDF file "VeryEnticingName_enc.pdf" exploits the Adobe Reader vulnerability and downloads a file in the current directory as "a.exe" then executes it. Now that we can execute arbitrary code on the system, we are free to do anything malicious.

Additional Threats

SF-EICAR performs the same "malicious" actions at each stage of the attack:

1. Download the encoded malware (DownloadedMalware.exe) and execute it. In some tests/ demos we set the disposition of the SHA-256 of this binary to malicious, so it may not run but the AMP for Endpoints Connector will give an alert.
2. Download the same malware but this time we modify the file randomly to produce a different SHA-256 for each run.
3. Download an encoded EICAR test file and copy it to the desktop.
4. Copy EICAR and append random bytes to make the SHA-256 unique then save the new file on the desktop.

For demonstration purposes, SF-EICAR displays a dialog box upon completion of each step.

Having completed the download of EICAR and download and execution of Win32.DemoMal.Rat.Client, the bootstrapper then performs process hollowing using one of the following randomly chosen applications:

- Svchost.exe
- Calc.exe
- Notepad.exe
- Certreq.exe
- Nslookup.exe
- Regsvr32.exe
- Regedt.exe
- Hostname.exe
- Chkdsk.exe

The injected code does the same “malicious” actions described above.

Next, it detects the architecture of the target system then downloads the correct binary for the platform. The downloaded binary, MalDemoDownloader.exe, which is XOR encoded, is decoded and executed. Next, the bootstrapper shows a message that it is done and exits after deleting itself from the disk.

MalDemoDownloader.exe does the same set of “malicious” activities and then downloads an encoded DLL, which is appropriate for the target platform. The DLL is decoded and then injected into one of the following randomly chosen processes:

- Spoolsv.exe
- Taskhost.exe
- Dwn.exe
- Explorer.exe

The injected DLL again does the same “malicious” activities discussed before, and downloads and executes another “malicious” file that displays a message box showing where it has been downloaded and that it is able to execute.

CHAPTER 3
DETECTION AND REMEDIATION

At this stage, two of the malicious files (Win32.DemoMal.Keylogger and Win32.DemoMal.Rat.Client) may have been detected and quarantined by desktop antivirus software. This neutralizes the immediate threat, but doesn't tell you how the malware gained entry into the enterprise, possibly allowing future malware to install itself in the same fashion. It is also impossible to tell from an antivirus product detection if the malware was detected prior to execution or after. You also don't know at this stage if other threats were introduced but not detected because antivirus products don't provide any upstream or downstream visibility. AMP for Endpoints allows you to trace back and discover malware that your antivirus software missed.

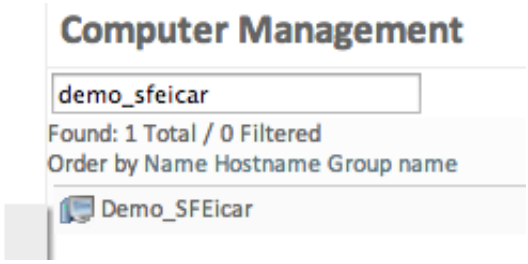
Using the AMP for Endpoints Console you can trace through this information to find out how the threats were introduced and check if any other computers saw the same threats. You can then use Outbreak Control lists to help prevent further infections.

Tracing Backwards

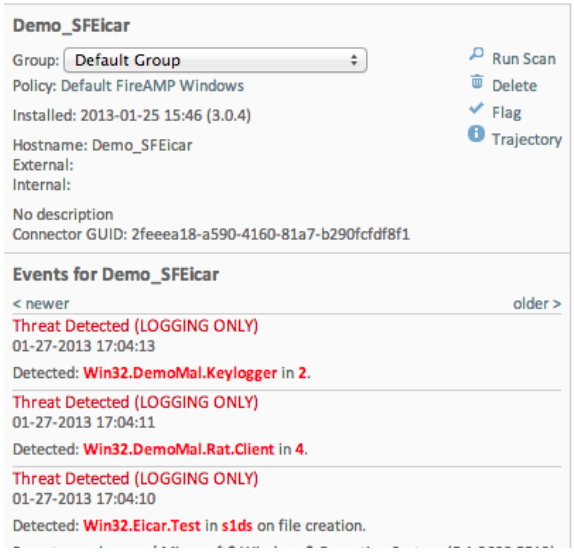
To trace an infection back to its origins you can use the Device Trajectory view in AMP for Endpoints. Device Trajectory can be accessed from different locations including the Dashboard, the expanded view of a device in the Computer Management screen, the expanded view of an event in the Events screen, and the Trajectory screen. This allows you to easily access detailed information about the device no matter how you discover the problem.

In this case we could assume the infected user calls the help desk after noticing suspicious activity and receiving notifications from their antivirus software. We have the user's computer name so we navigate to the Computer Management screen by selecting Computers under the

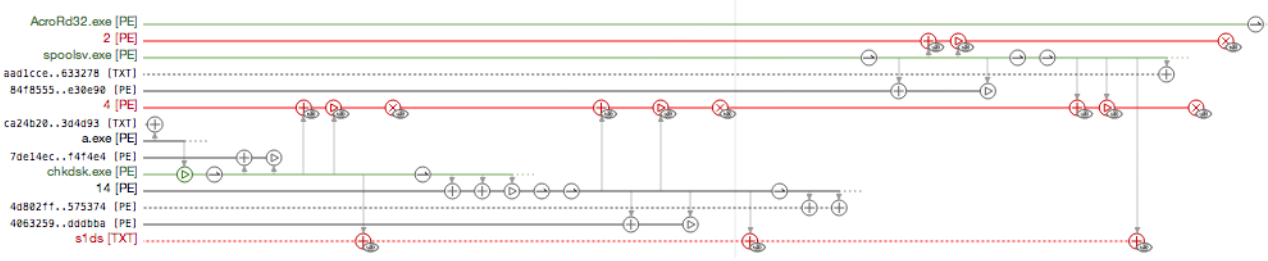
Management menu. If there are a large number of computers in your deployment, enter the computer name in the search box.



When we expand the computer view for Demo_SFEicar we can see numerous threat detection event entries but we don't know if they are related and how they gained entry or if they managed to download and install any other threats that were not immediately detected.

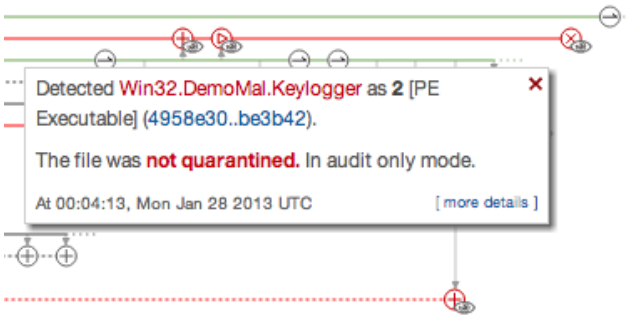


To trace the infections back to their source and see if other suspicious files or network activity took place we can launch the Device Trajectory for this computer by clicking on the Trajectory link in the list at the top right. When Device Trajectory is first launched it shows us a view of the most recent events on the computer. In this case we immediately see three files with a malicious disposition (shown in red).

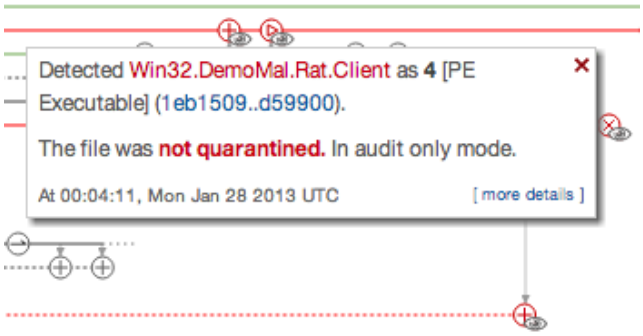


Additionally, there is an Indication of Compromise event in the Trajectory that means a particular set of suspicious events occurred that were flagged by AMP for Endpoints. In this case the compromise event was triggered by Adobe Reader creating and executing an executable file, which is often a sign that a PDF file containing an exploit was opened.

Going back to our malicious files in the trajectory, the most recent event type associated with two of the malicious files are detections. The most recent is a detection of Win32.DemoMal.Keylogger, which matches what was observed in the events view of this computer.



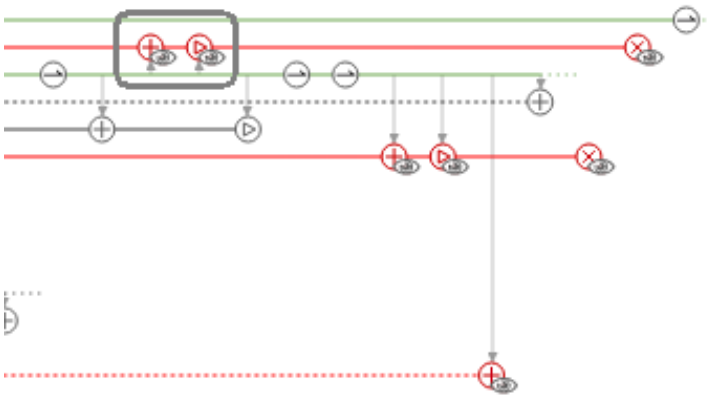
The other detection is for Win32.DemoMal.RAT.Client, the other detection observed in the events view.



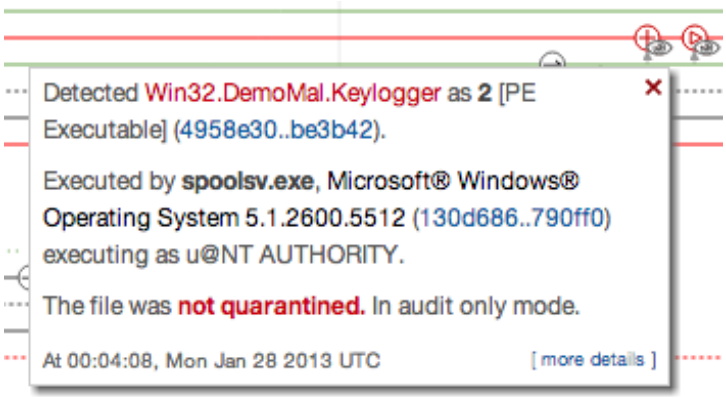
You'll notice that neither file was quarantined. This is because we ran the demo in audit-only mode to show the full range of actions malicious files could take and how each action is recorded by AMP for Endpoints.

Having both a keylogger and a RAT (remote access Trojan) on a computer can have serious implications. Not only do we want to know how the threats were introduced, but we will also want to check for any upstream and downstream activity that could indicate data was exfiltrated.

First we'll trace back the activity and origin of Win32.DemoMal.Keylogger, in this case a portable executable file with the name "2". Shortly before the detection event, we can see creation and execution events for this file.

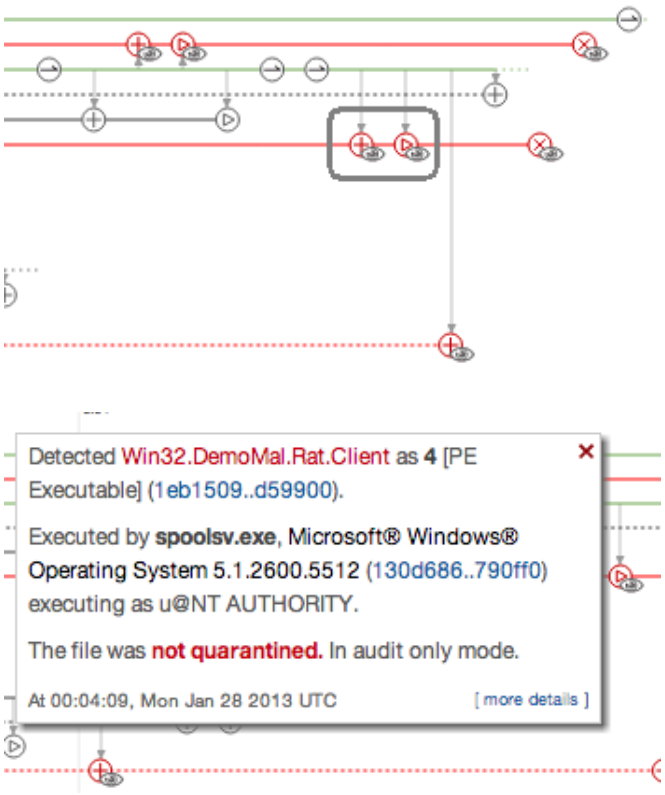


Clicking on the events for more details shows that the file was created then executed by spoolsv.exe, the Windows print spooler service.

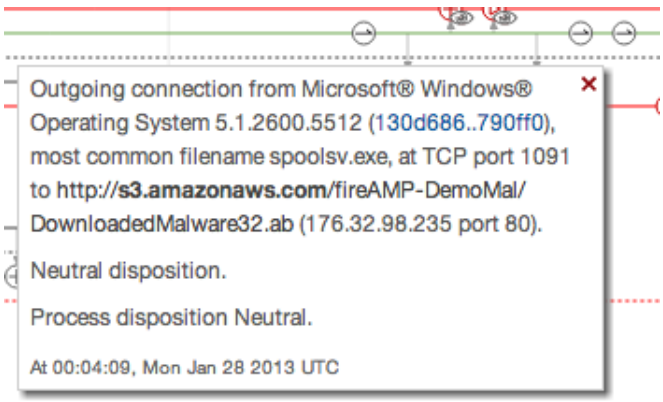


Normally this service should never create portable executable files or execute them. In some cases a malicious file could name itself spoolsv.exe to try to seem innocuous, but in this case we see that it has a clean disposition (green) meaning that it is the legitimate Windows utility.

Upon further examination, we see that spoolsv.exe also created and executed Win32.DemoMal.Rat.Client.



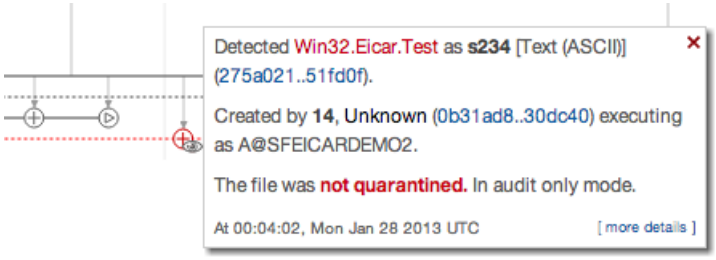
From the Device Trajectory we can also see that spoolsv.exe created a malicious text file (in this case the EICAR test file). In addition to creating and executing files, spoolsv.exe made outbound network connections to a remote host.



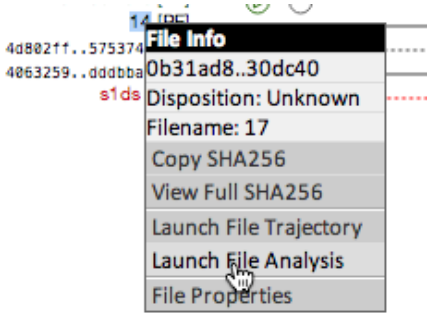
This is definitely not behavior you would expect to see from the print spooler service, which likely indicates some sort of DLL injection has occurred. This means that there would have

been some sort of earlier malicious activity on the computer that initiated the DLL injection so we need to keep tracing backward.

Looking further back in time we can see that a copy of the EICAR text file was also created by a portable executable file named “14”.

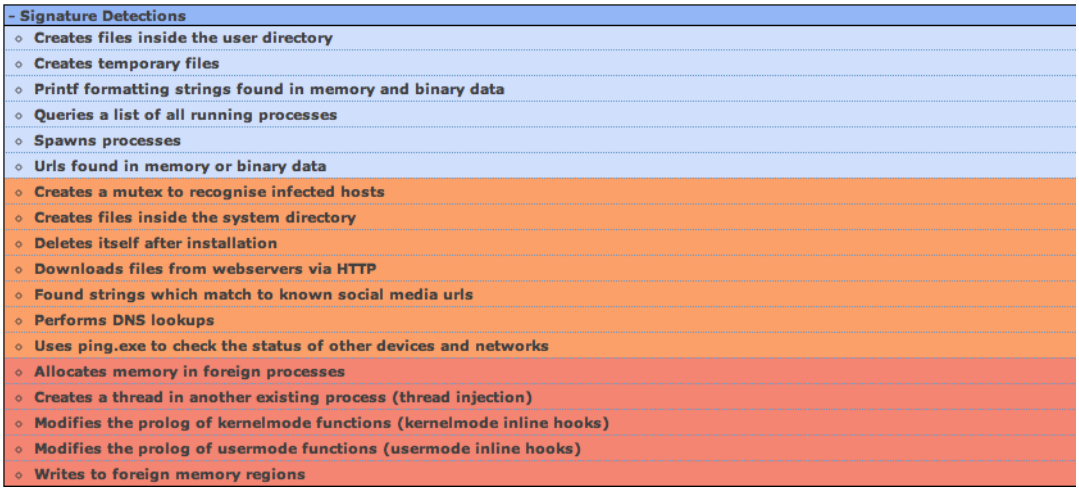


This file has an unknown disposition, meaning that it has likely not been seen before so it has not yet been determined to be either clean or malicious. To find out more about this file, we can right click on its name in the left column of the Device Trajectory and launch File Analysis.

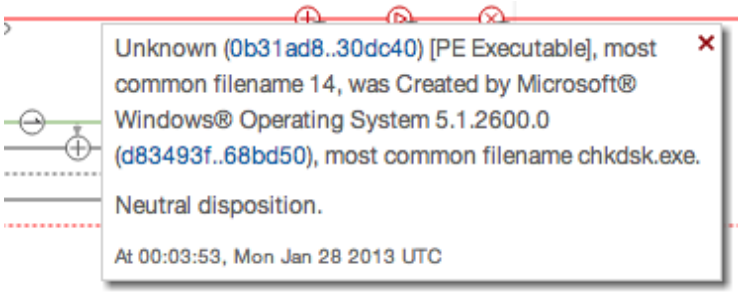


File Analysis will attempt to execute the file in a sandbox environment and report back key features about it that you can use as decision support. For this file we’ll start by looking at the Classification / Threat Score section to get a sense of what it does.

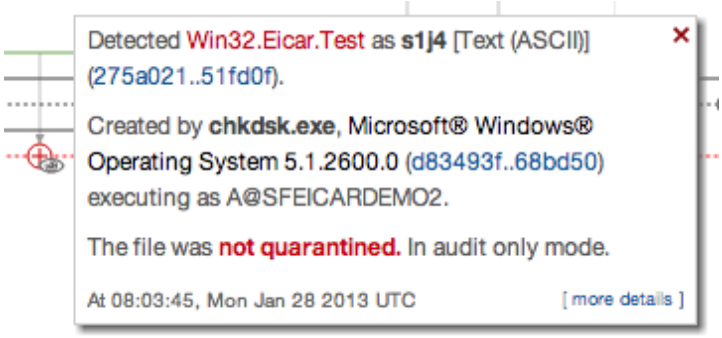
We can see that there are signs of malicious activity coming from this executable, so now let's look at the Signature Detections section to get a better sense of what it's trying to do.



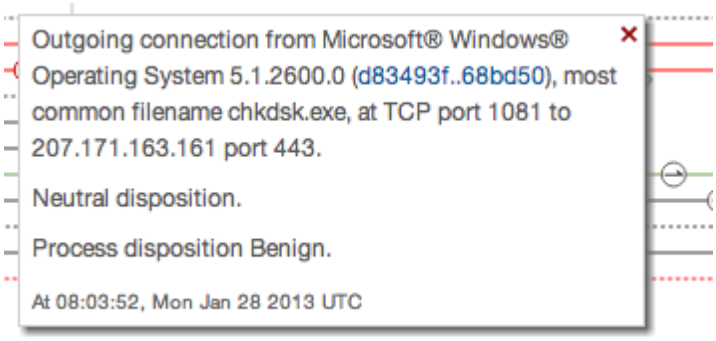
There is suspicious network activity like downloading files and using ping.exe, but the red activities like thread injection tell us that this executable is trying to hide itself by running inside another process. Going back to Device Trajectory we can see that the parent of this file is chkdsk.exe, the Windows check disk utility.



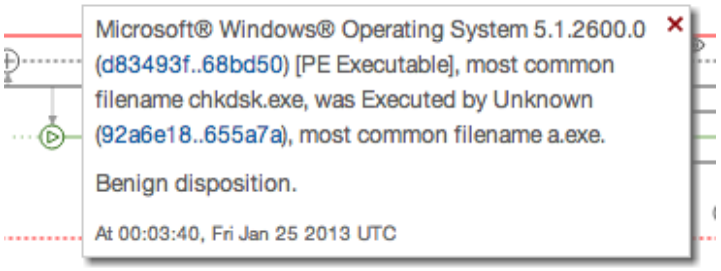
Like spoolsv.exe, we know that this instance of chkdsk.exe is the legitimate Windows utility because it has a clean disposition. It is likely that there is some form of process injection has occurred. Other suspicious activity helps to confirm this. We can see that chkdsk.exe created a copy of the same malicious text file that spoolsv.exe did.



We also see that chkdsk.exe made multiple outbound network connections, which is something that a disk utility should never do.

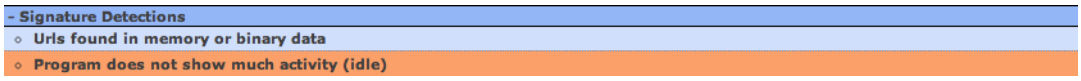


Continuing to trace backwards we find that this instance of chkdsk.exe was executed by an unknown file called a.exe.



Since a.exe is an unknown file, let's launch File Analysis to see what it tells us about it. The Classification / Threat Score reveals little in this case, showing only slightly suspicious activity.

The Signature Detections show very little at first glance.



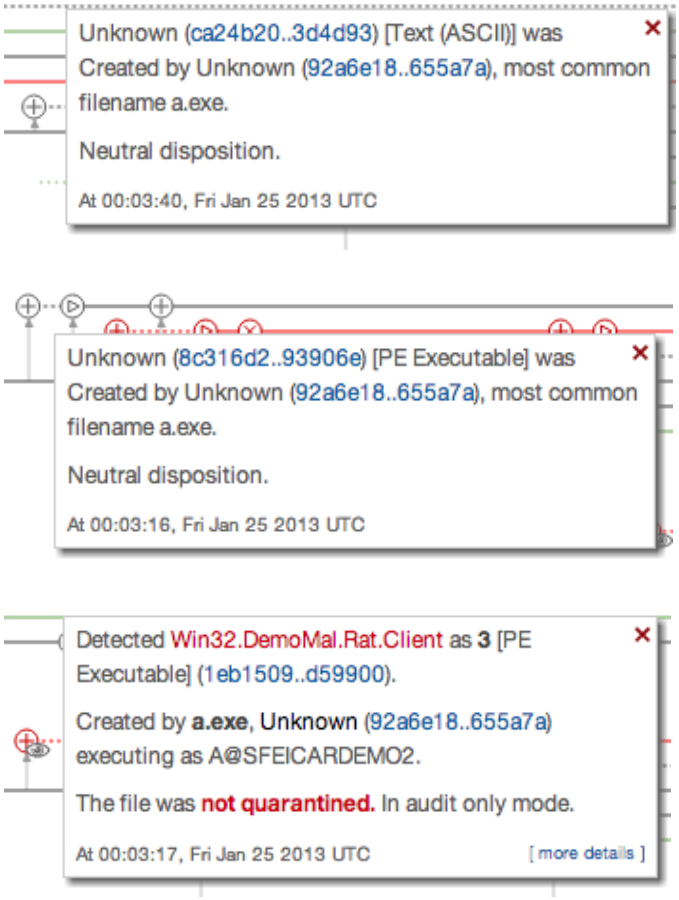
The executable doesn't show much activity while it's running and contains URL strings. This is typical of a downloader - a small application whose sole purpose is to execute on a computer undetected then download additional threats. Looking at the URLs list in the String Analysis section, we see five different URLs.

- URLs

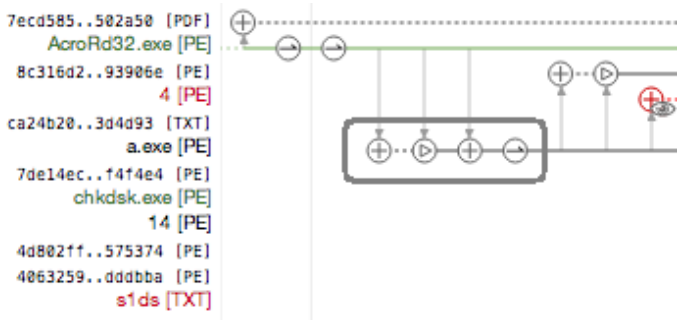
String value	Source
http://s3.amazonaws.com/fireamp-demomal/downloadedmalware32.ab	0047343702
http://s3.amazonaws.com/fireamp-demomal/downloadedmalware64.ab	0047343702
http://s3.amazonaws.com/fireamp-demomal/eicarencoded.ab	0047343702
https://s3.amazonaws.com/fireamp-demomal/maldemodownloader32.ab	0047343702
https://s3.amazonaws.com/fireamp-demomal/maldemodownloader64.ab	0047343702

We now get a sense that this file is suspicious, but there may not be enough evidence yet to convict this file. However, going back to the Device Trajectory and tracing through the actions

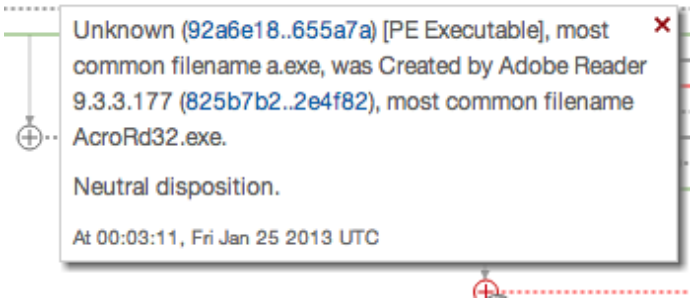
of a.exe provides us with more proof that this file is malicious. We see that a.exe creates three files on the computer and executes two of them.



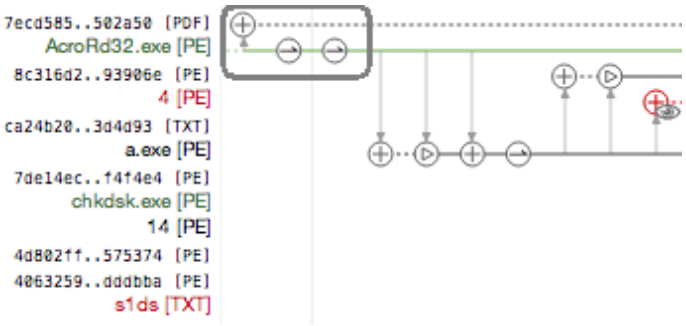
Of particular concern is that two of the files are unknown and the third is detected as a malicious file, namely the Win32.DemoMal.Rat.Client Trojan. This confirms our suspicion that a.exe is a downloader and possibly the origin of our system compromise. Now we want to trace this file back to find how it was introduced. Scrolling back in our Device Trajectory we locate the initial creation and execution of a.exe followed by an outbound network connection.



Looking at the details of the initial file creation event for a.exe we observe that it was created by AcroRd32.exe - Adobe Reader.

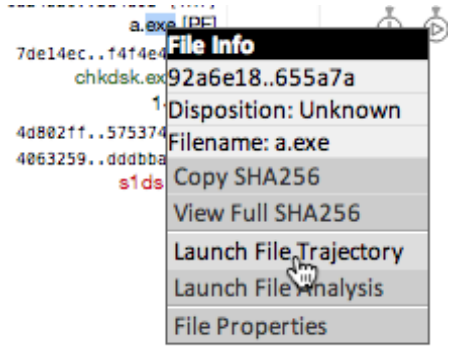


As with spoolsv.exe and chkdsk.exe, Adobe Reader should never create and execute files. Looking at the Device Trajectory we see that Adobe Reader was used to open a PDF file then immediately made outbound network connections before creating a.exe.



This leads us to believe that the PDF file that was opened may have exploited a vulnerability in Adobe Reader in order to download and execute a.exe since this is the earliest suspicious event in the Device Trajectory for this computer.

The next step is to make sure that no other computers in our AMP for Endpoints deployment have been compromised by this threat. First, we right-click on the a.exe file in Device Trajectory to launch File Trajectory.

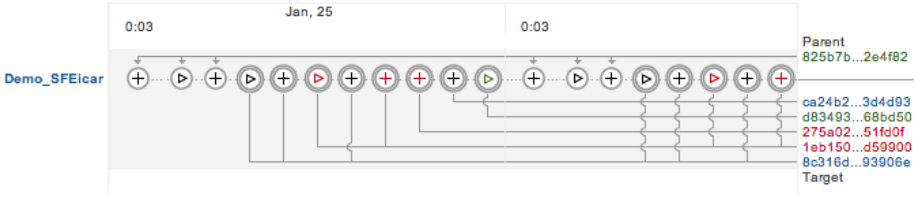


Where Device Trajectory gives us an in-depth view of all file activity on a single computer, File Trajectory shows us the presence of a file and its activity across all AMP for Endpoints Connectors in our deployment. Looking at the Entry Point in File Trajectory, we see our compromised computer from Device Trajectory listed.

This tells us that Demo_SFEicar was the first AMP for Endpoints Connector to observe a.exe in our deployment, often referred to as “patient zero”. Confirming what we saw in Device Trajectory, Adobe Reader 9.3 is listed as the parent file for a.exe.

The only computer listed in the Trajectory field is Demo_SFEicar. This means that it was the only computer in our deployment to observe the file, which could be indicative of a targeted attack.

We can also expand the Trajectory view for a.exe by clicking on the computer name. This shows the parent file (AcroRd32.exe) as well as all target files it created or acted on.



Remediation

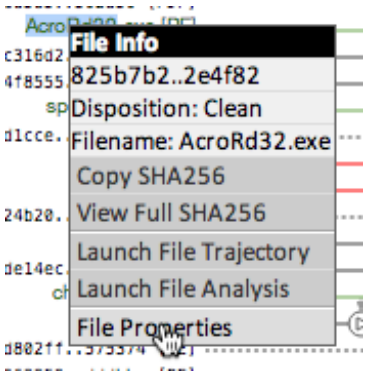
Now that the origin of the compromise has been identified we can begin the process of remediation and blocking future compromise by this threat. Using Outbreak Control lists we can stop vulnerable applications from executing, detect and quarantine unknown files, and black list associated IP addresses.

Application Blocking

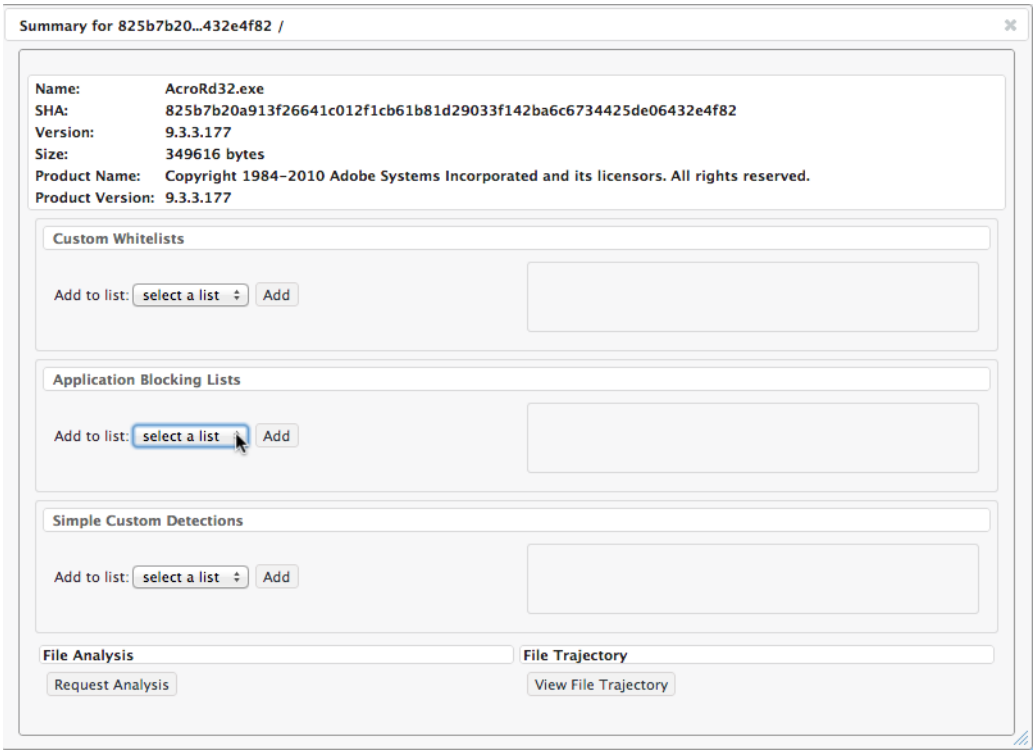
First, we can close off the original entry point of the infection by upgrading Adobe Reader to a version that is not vulnerable to the exploit used. We can also prevent the vulnerable version

from running by adding it to an Application Blocking List. This is convenient if you don't want users running the vulnerable application before the upgrade is deployed or in cases where there is no patch or fix available yet.

To add AcroRd32.exe to an Application Blocking list, right click on its SHA-256 anywhere from Device Trajectory or File Trajectory and select File Properties from the context menu.



From the File Properties dialog, select the name of a list from the Application Blocking Lists section and click Add.

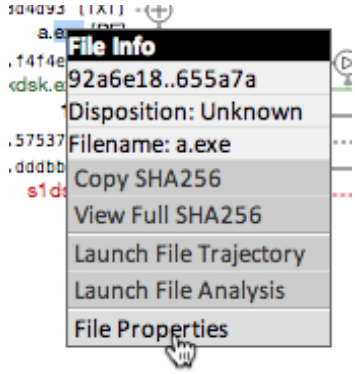


Make sure that the Application Blocking List you have selected is applied to any applicable policies. You can repeat this step to add AcroRd32.exe to other Application Blocking Lists as

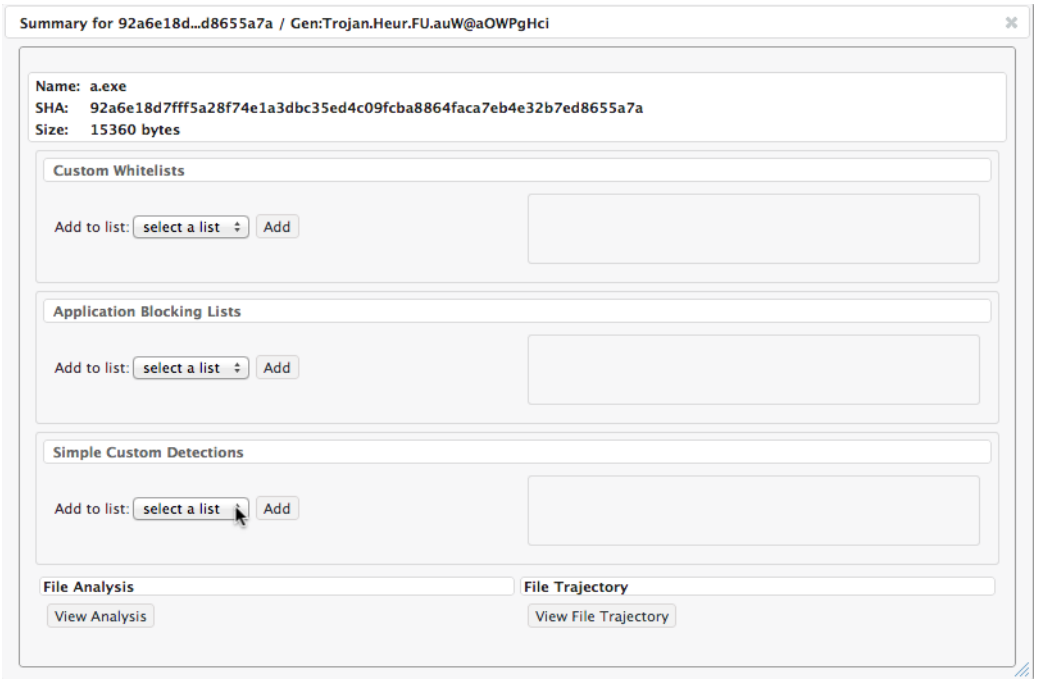
well. An Application Blocking List does not quarantine applications you add to it, but instead prevents them from running.

Simple Custom Detections

Next we'll add a.exe to a Simple Custom Detection list. This will not only detect and quarantine any new instances of the file, but any existing ones as well the next time they are created, moved, or executed. We can do this the same way we added Adobe Reader to the Application Blocking List. In Device Trajectory, right click on a.exe in the left column and select File Properties from the context menu.

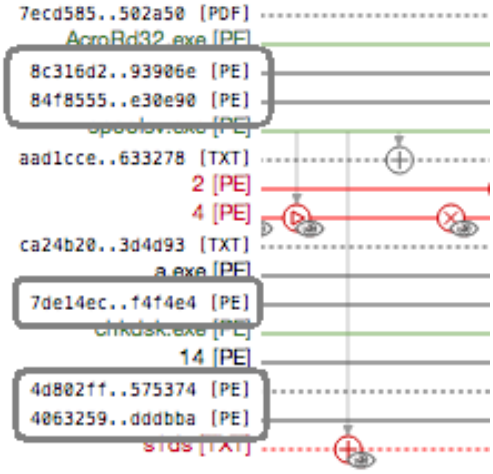


From the File Properties dialog, select the name of a Simple Custom Detection list and click Add.



You can do this multiple times for each Simple Custom Detection list you want to add the file to.

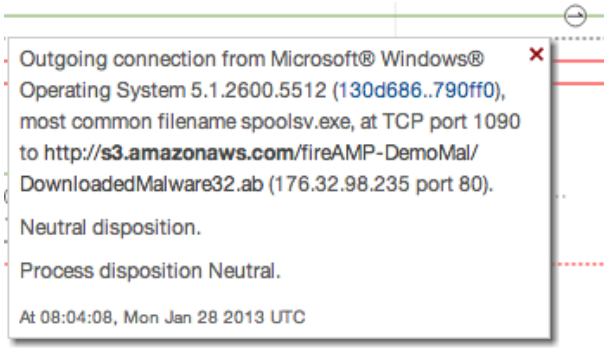
Now you can begin adding other unknown files created by this threat to Simple Custom Detection lists as well by going through the Device Trajectory or File Trajectory and repeating the above procedure for these files.



IP Black Lists

The new Device Flow Control (DFC) feature of AMP for Endpoints Connector version 3.1.0 and later allows you to monitor and block network connections. You can create custom IP White and Black lists and assign them to policies wherever they're needed. In this example we have multiple threats downloading files from remote hosts. The remote access Trojan component of our demo threat could also send and receive information through a network channel in a real world scenario. DFC allows you to block both upstream and downstream connections associated with a threat.

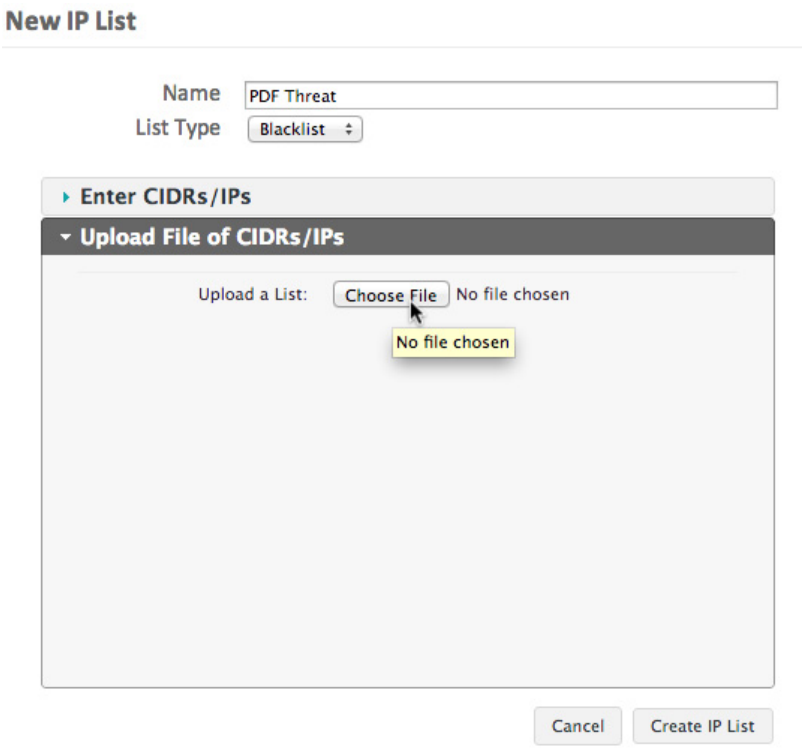
In our scenario we can go through the Device Trajectory looking for network traffic.



Copy the relevant IP addresses and paste them into a text file. You can also add specific ports where necessary (eg. x.x.x.x:yy). You can also add entire CIDR blocks to your IP lists if you choose.

WARNING! Exercise caution when adding IP addresses or CIDR blocks to a black list. Malware may often be hosted on addresses also hosting legitimate websites and services.

Once you have added all the IP addresses you want to your list, go to Outbreak Control > IP Black/White Lists in the AMP for Endpoints Console. Click on Create IP List then give the list a name, select Blacklist for the List Type, and choose the Upload File of CIDRs/IPs. Click Choose File and select your text file from the dialog.



Once your list has been uploaded, click Create IP List. Next, go to any policies you want to add your list to and edit them.