



# AMP for Endpoints ZAccess

**Last Updated:** August 9, 2016



# CHAPTER 1

## INTRODUCTION

This attack scenario replicates an “in the wild” infection of ZeroAccess. In addition to being a rootkit, ZeroAccess (aka ZAccess, Sirefef, Max++) generates advertising revenue for the attackers through click fraud.

ZeroAccess is often installed on computers via drive-by download, often from websites that have been compromised with the Blackhole or Nuclear Pack toolkit. This toolkit attempts to exploit several browser vulnerabilities when a user visits the compromised website. In this scenario we will use AMP for Endpoints to discover the malicious activity and find all associated secondary infections.

---

**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 visits a compromised website that exploits CVE-2013-0422, the Java 7 Security Manager Bypass, a 0-day vulnerability that was exploited in the wild in January 2013. Like many attacks, the exploit sets off a chain of events including downloading and executing 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 compromise a legitimate website and use it to host the exploit or set up a server that hosts the exploit and attempt to lure users by placing a link to it in email messages or on social networking sites.




When the user visits the site, it exploits the Java vulnerability and downloads ZeroAccess then executes it. ZeroAccess makes a network connection to a remote site to get geographical information, then in some cases starts an Adobe Flash Player install to escalate its privileges.

# CHAPTER 3

## DETECTION AND REMEDIATION

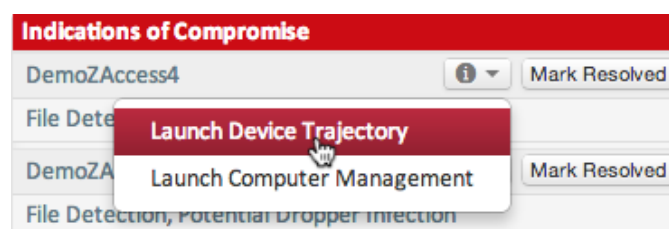
When you log in to the AMP for Endpoints Console the first page you see is the Dashboard Overview. This page shows you recent file and network detection events from your AMP for Endpoints Connectors. It's a convenient summary of the major trouble spots in your AMP for Endpoints deployment that allows you to perform triage to determine which computers are in most need of immediate attention.

The Indications of Compromise on the Dashboard Overview helps with triage by listing computers with multiple events or separate events that correlate with certain types of infections. In our scenario we see that the top computers with indications of compromise have experienced file detections as well as events typically associated with a dropper infection. A dropper infection occurs when a single file repeatedly attempts to download malware onto a computer.

Indications of Compromise		
DemoZAccess4		Mark Resolved
File Detection, Potential Dropper Infection		
DemoZAccess2		Mark Resolved
File Detection, Potential Dropper Infection		
DemoZAccess5		Mark Resolved
File Detection, Potential Dropper Infection		
DemoZAccess6		Mark Resolved
Potential Dropper Infection, File Detection		

Since computers at the top of the list are considered to have more severe compromise indicators than those lower on the list, we'll start at the top. Click the information icon next to

the computer name in the list and select Device Trajectory to begin the incident response process.

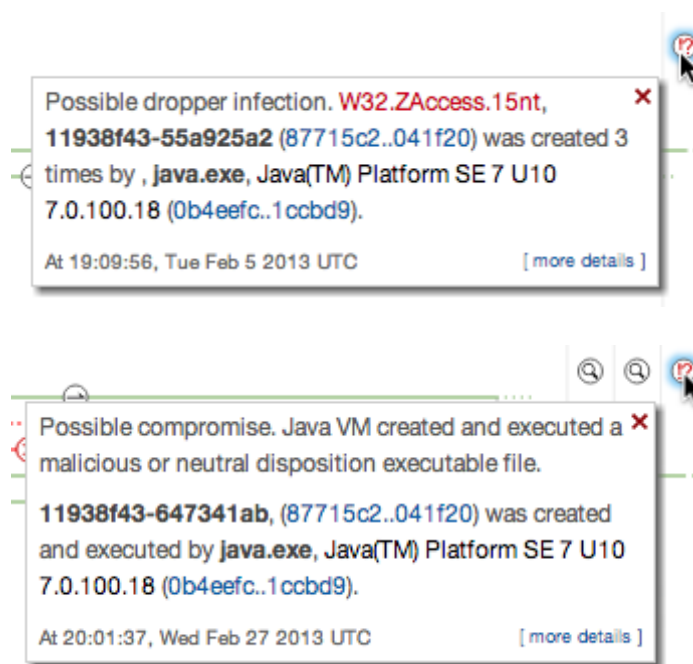


## Tracing Backwards

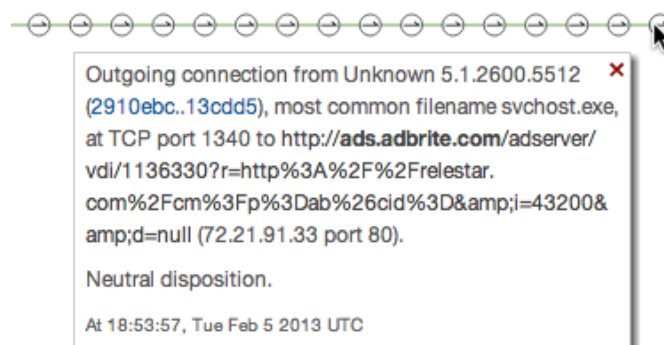
When we first look at the Device Trajectory for this computer, we immediately see obvious signs that it has been compromised since there are four red entries in the file list on the left, indicating known malware detections.



The most recent events in the trajectory - those furthest to the right - are also suspicious. First, we see two Indication of Compromise events showing a potential dropper infection and a possible Java compromise.



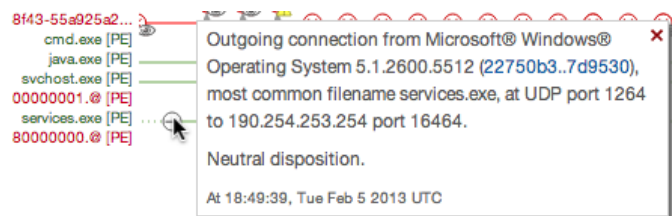
There are also a number of network connection events. Those events show that svchost.exe (Windows Service Host), a known clean application, is making outbound network connections. While a clean application making outbound network connections is not suspect on its own, svchost.exe does not normally exhibit this behavior. If we click on some of the network connections we see some interesting information in the details.



It appears that the outbound connections are to advertising sites that pay revenue each time someone clicks on the ad. This is often indicative of a form of click fraud where the malware author generates income by having compromised computers click multiple advertisements.

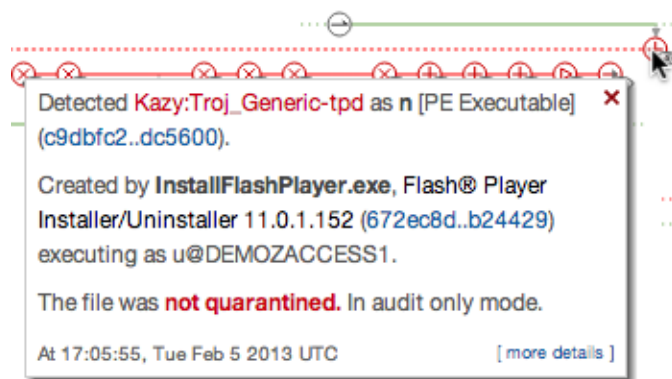
Since this activity is being performed by svchost.exe it's likely that some other process has injected itself into its process to evade detection.

Looking further back we also see that services.exe (Windows Service Control Manager) has moved two files with a malicious disposition. This is unusual activity for services.exe, so this tells us something else is going on with the process. Further back in the time line we see that it made an outbound network connection, which is also unusual behavior.



The connection on a high UDP port is the command and control channel ZAccess uses as part of its back door. While Zero Access is a stealthy rootkit, we are still able to see most of its activity in Device Trajectory.

In some cases we will also see that Zero Access commences an install of Adobe Flash Player. This is actually used to elevate its privilege level on the computer in order to install a kernel-mode rootkit.



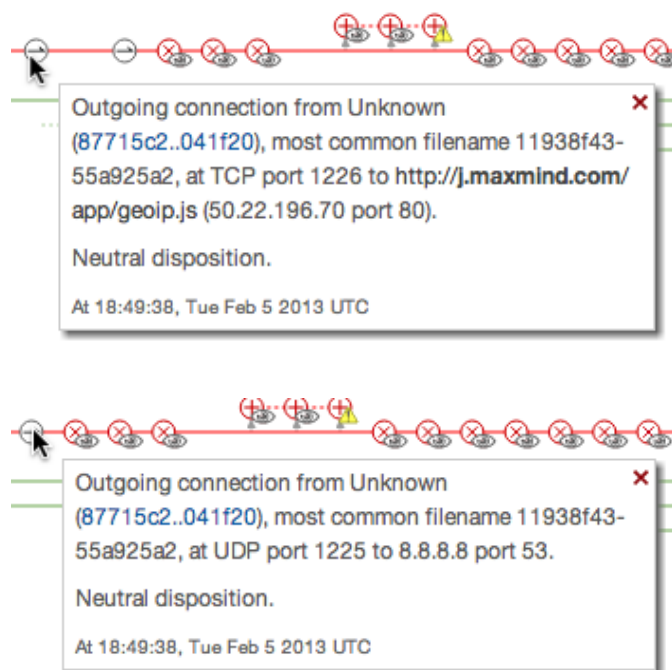
Now we'll look at the most obvious sign that this computer is compromised. We can see that there's a randomly-named file with multiple detection events as well as activity like creating other malware on the computer.



Immediately after the file was created and executed, it made two network connections - one to j.maxmind.com and one to a Google DNS server. These are both common behaviors of Zero

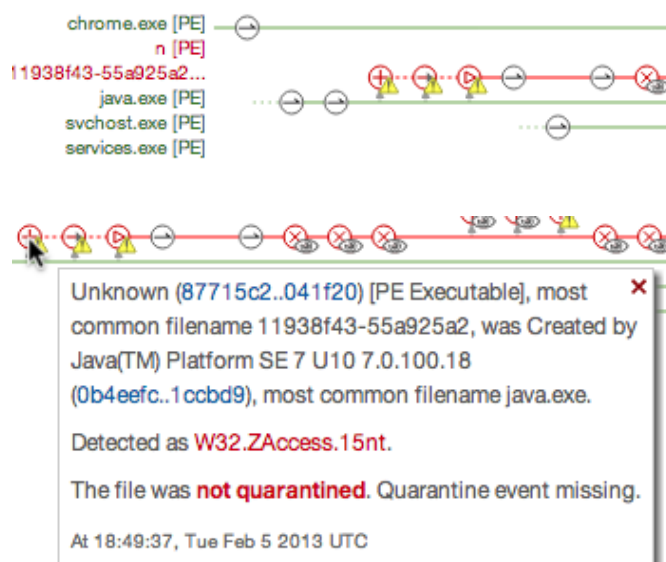


Access as it gathers geolocation information about the computer and ensures it can perform DNS resolution.

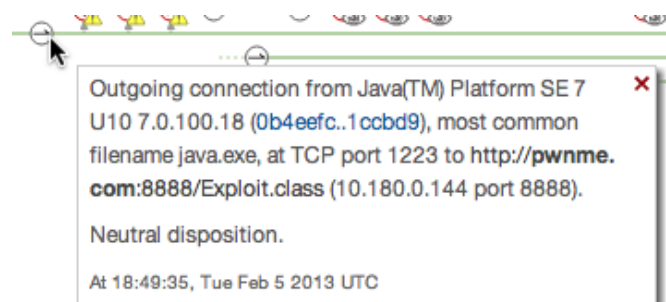


The randomly named file appears to be the first malicious file observed in the trajectory and is likely the root of the other infections. The question we want to answer now is how the infection

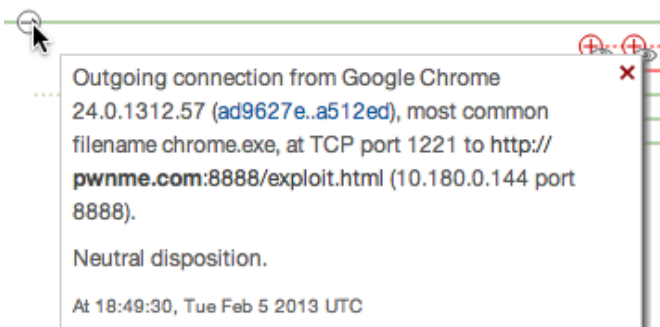
was introduced. The randomly named file appears to have been created, moved, and executed by java.exe (Oracle Java).



Java.exe makes two network connections just before creating the malware, so it's likely this is where our malware originates from.



Since Java does not normally create and execute portable executable (PE) files that it downloads from remote servers, it's possible that a vulnerability was exploited to cause this behavior, since chrome.exe (Google Chrome) can be seen visiting the same site.



So in summary, we determined that the user visited a website that exploited a vulnerability in Java to download and execute malware on the computer. That malware then creates other files and infects legitimate processes to hide its presence. Finally, it causes svchost.exe to make connections to various ad servers to generate revenue through click fraud.

Since Java is widely used it's a good idea to check if other computers in our AMP for Endpoints deployment have been exploited through this vulnerability as well. First, we navigate to Threat Root Cause under the Analysis menu. Here we see that java.exe is the top program introducing malware into our environment.

Section: Overview Summary Timeline						
Threat Root Cause - Summary						
This table is a detailed report of the top 10 applications that were implicated in introducing malware into your environment. Entry points for malware should be investigated and, if possible, remediated. To see the top threats introduced by a program click its name.						
Program	SHA-256	Threat Name	Version	Threats Introduced	Computers Affected	Event Type
java.exe	0b4eefc0...201ccbd9	n/a	7.0.100.18	10	6	60% created 20% executed 20% moved
a.exe	0723932d...1fbfe85f	n/a		6	2	33% created 33% executed 33% moved
explorer.exe	1e675cb7...885ef455	n/a	6.0.2900.5512	3	3	100% moved100% executed
zaccess.exe	87715c24...fb041f20	W32.Generic.Gen.15nt		2	6	100% created
InstallFlashPla...	672ec8dc...edb24429	n/a	11.0.1.152	1	3	100% created

Clicking on java.exe will expand the list of threats being introduced. We see that several computers have been affected and all of them were affected by the same malware - Zero Access. Next we'll launch File Trajectory to find further information on the threat.

Program	SHA-256	Threat Name
java.exe	0b4eefc0...201ccbd9	n/a

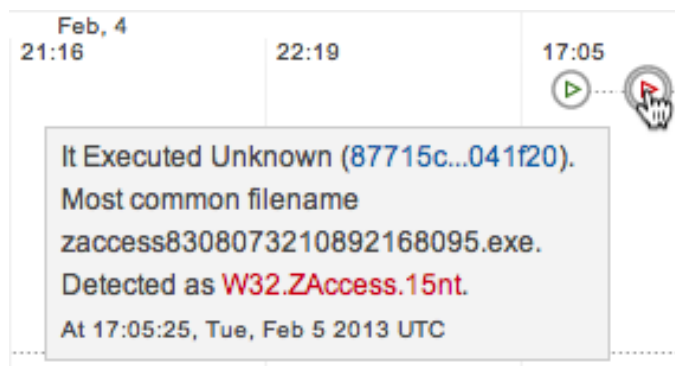
  

<b>File Info</b>	
0b4eefc0...201ccbd9	
Disposition: Clean	
Filename: java.exe	87715
Copy SHA256	87715
View Full SHA256	87715
Launch File Trajectory	87715
Launch File Analysis	87715
File Properties	87715

We can immediately see that multiple computers have had threats created and executed by java.exe.



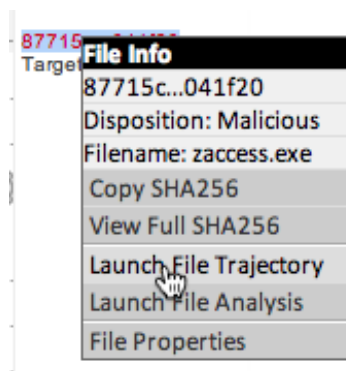
Looking at the details we can see that in all cases the malware introduced was ZAccess.



Now we want to know more about this infection from a full network perspective so we first expand one of the trajectories by clicking on a computer name.



Then right-click the ZAccess SHA-256 value on the right and select File Trajectory from the context menu to see its trajectory.



We can see the first time that Zero Access was observed on our network and the last time along with the total number of times it was observed.

Visibility	your network	community
First Seen	February 4, 2013 at 21:32	
Last Seen	February 5, 2013 at 20:40	
Observations	21 (as target), 14 (as source)	0 (as target), 0 (as source)

This tells us how long computers have been compromised and particularly in the case of malware that can steal information, how much time we were exposed for. We can also see the first computer that saw the threat, or “patient zero”.

Entry Point
First Seen On <a href="#">DemoData / ZAccess / Demo_ZAccess1</a>

In the case of a virus or worm this information would be most useful, but in our scenario we would likely go to the user of that computer to find out if they received an email or instant message containing a link that they then sent to the other users. This information combined with the trajectory information can tell us how the malware affected more than one computer and allow us to take further steps at the network perimeter or on mail and messaging servers.

Looking at the trajectory for ZAccess we see the other files it created and launched.



We can further drill down on the computers to check the other files ZAccess introduced.



We can continue to right-click on the various SHA-256s associated with the threat to make sure we haven't missed anything in the Device Trajectory. In this case, we have covered the entire trajectory of the threat on a device as well as across our network.

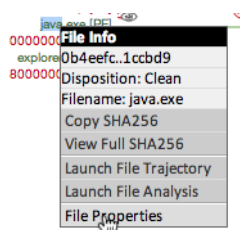
## Remediation

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.

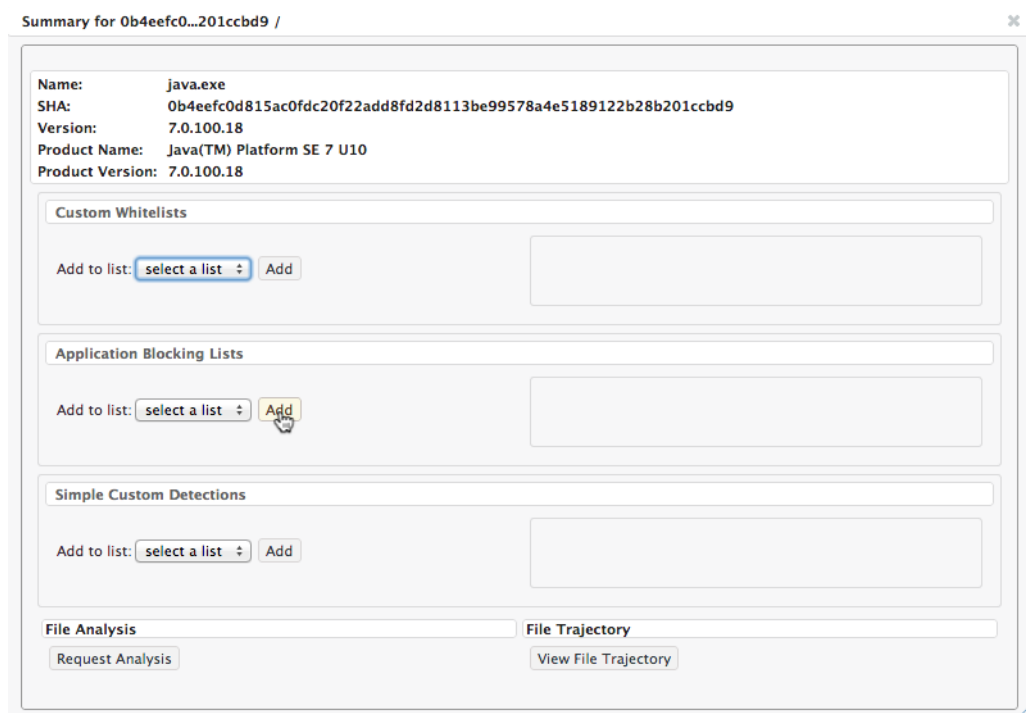
### Application Blocking

First, we can close off the original entry point of the infection by upgrading Oracle Java 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 java.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.

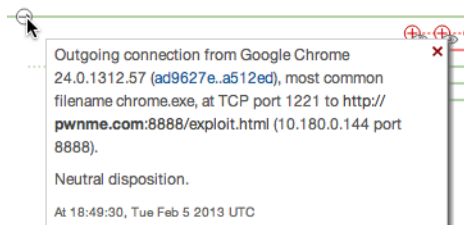


Make sure that the Application Blocking List you have selected is applied to any applicable policies. You can repeat this step to add AcroRd32.exe to other Application Blocking Lists as well. An Application Blocking List does not quarantine applications you add to it, but instead prevents them from running.

## 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.

#### New IP List

Name

List Type

Blacklist

Enter CIDRs/IPs

Upload File of CIDRs/IPs

Upload a List:

Choose File

No file chosen

No file chosen

Cancel

Create IP List



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.