



# Automatic Analysis of Low Prevalence Files, Resulting in a Retrospective Quarantine

May, 2016

## CONTENTS

<b>PREFACE .....</b>	<b>2</b>
<b>1.0 Introduction .....</b>	<b>3</b>
<b>2.0 The Attack.....</b>	<b>3</b>
<b>3.0 Detection and Remediation.....</b>	<b>3</b>
3.1 Tracing Back .....	4
3.2 Retrospective Detection & Remediation Due to Low Prevalence .....	6
3.2 How it Started.....	9
<b>4.0 Summary .....</b>	<b>11</b>

## PREFACE

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS DOCUMENT ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS OR INFORMATION.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPS WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

All contents are Copyright © 2016 Cisco Systems, Inc. and/or its affiliates. All rights reserved.

## 1.0 Introduction

The following scenario describes an encounter with a previously unknown malware variant infection in the wild, in which FireAMP detects a suspicious filename being used by an executed file. This file is submitted to Threat Grid for further inspection due to its low prevalence within the organization. Threat Grid determines that the file is malicious due to its activity during dynamic analysis, and it is quarantined using FireAMP's Cloud Recall technology. FireAMP is then used to trace the attack back to its initial infection vector using Device Trajectory.

## 2.0 The Attack

The attack is a simple yet effective masquerade of an executable file that is made to appear as a benign PDF document that is downloaded by a user from a Web browser. This is done by adding “.pdf” to the file’s “.exe” extension, resulting in the following: “.pdf.exe”. Microsoft Windows will hide known file extensions by default, including “.exe”, so to the untrained eye the file will appear to have a “.pdf” extension.

The masquerader can take this one step further by changing the icon of the executable to that of a PDF document when the default handler is a popular PDF reader, such as Adobe Acrobat.

## 3.0 Detection and Remediation

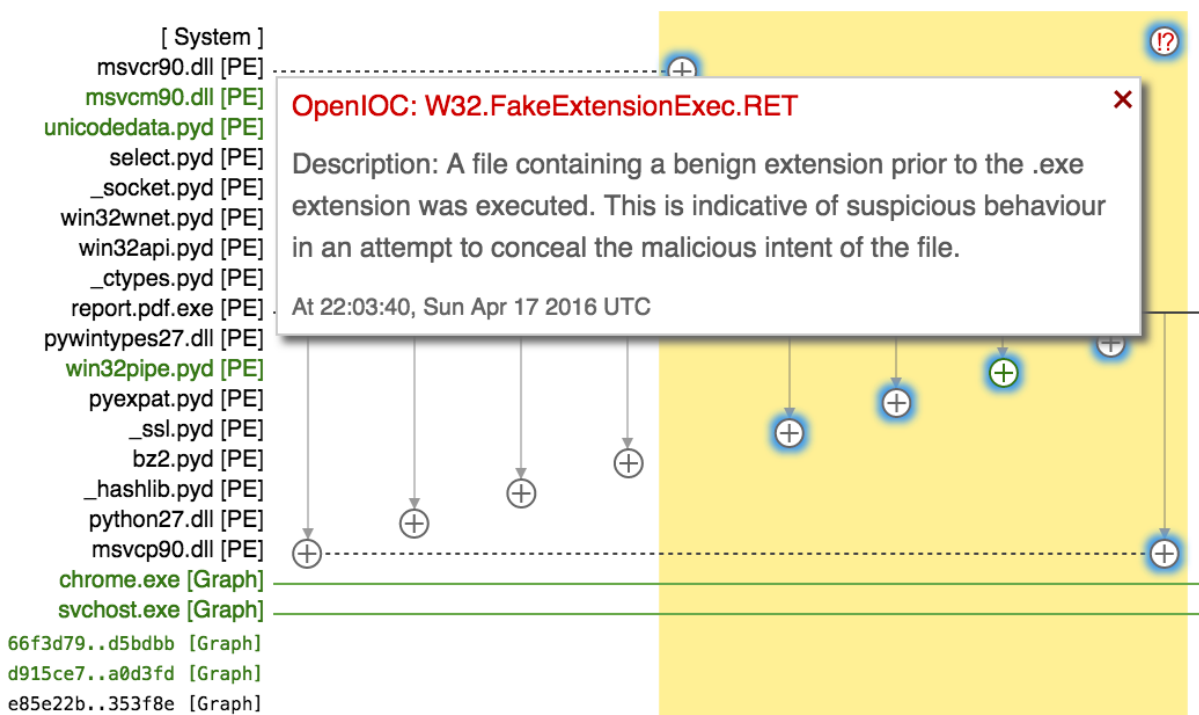
The first page you see after logging into the FireAMP Console is the Dashboard Overview. This page displays recent file and network detection events from your FireAMP Connectors. It's a convenient summary of the major trouble spots in your FireAMP deployment, which allows you to perform triage to determine which computers are in most need of immediate attention.

The **Indications of Compromise** section on the Dashboard Overview helps with triage by listing computers with multiple events, or with separate events that correlate with certain types of infections. In our scenario, we see that the top computers with indications of compromise have experienced Generic IOC detections.

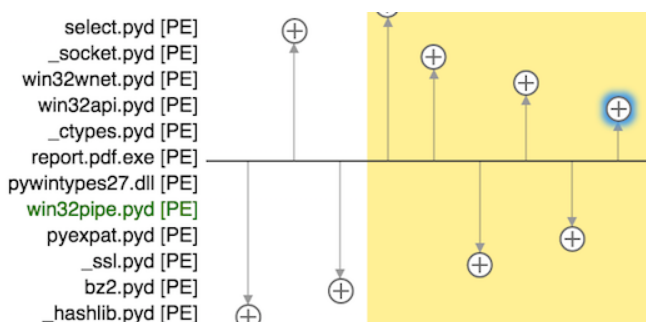
Since computers at the top of the list are considered to have more severe compromise indicators than those lower down on the list, we start at the top. Click the information icon next to the computer name in the list, and select **Device Trajectory** to begin the incident response process.

## 3.1 Tracing Back

Upon opening the Device Trajectory for one of the Generic IOC Detections we see an Indication of Compromise due to a file executing with a known extension pattern being used to masquerade executables as benign documents: *report.pdf.exe*:

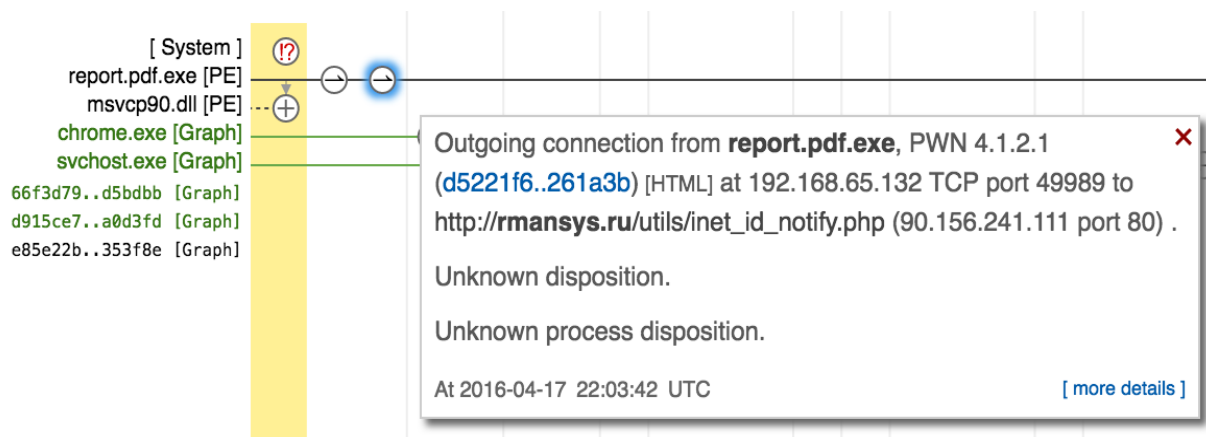
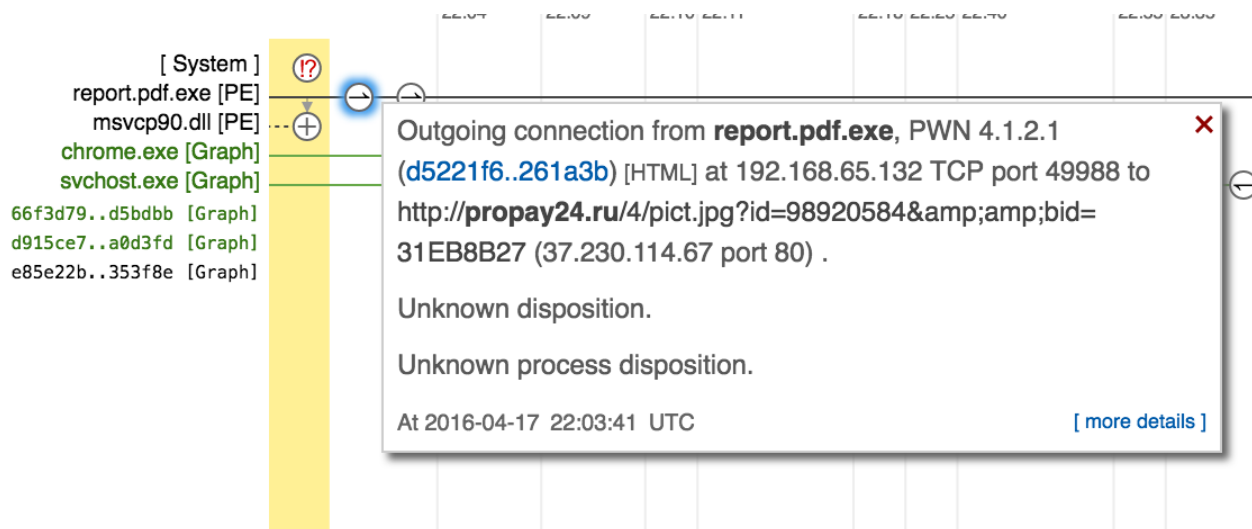


The event is coming from the *report.pdf.exe* executable that appears to be performing a number of actions. These include the creation of a number of “.pyd” files, which could indicate that this executable is packaged with a Python interpreter such as [py2exe](#):



This does not inherently signify that this file is malicious, but it may assist in the post-compromise investigation process when analyzing the binary for malicious intent.

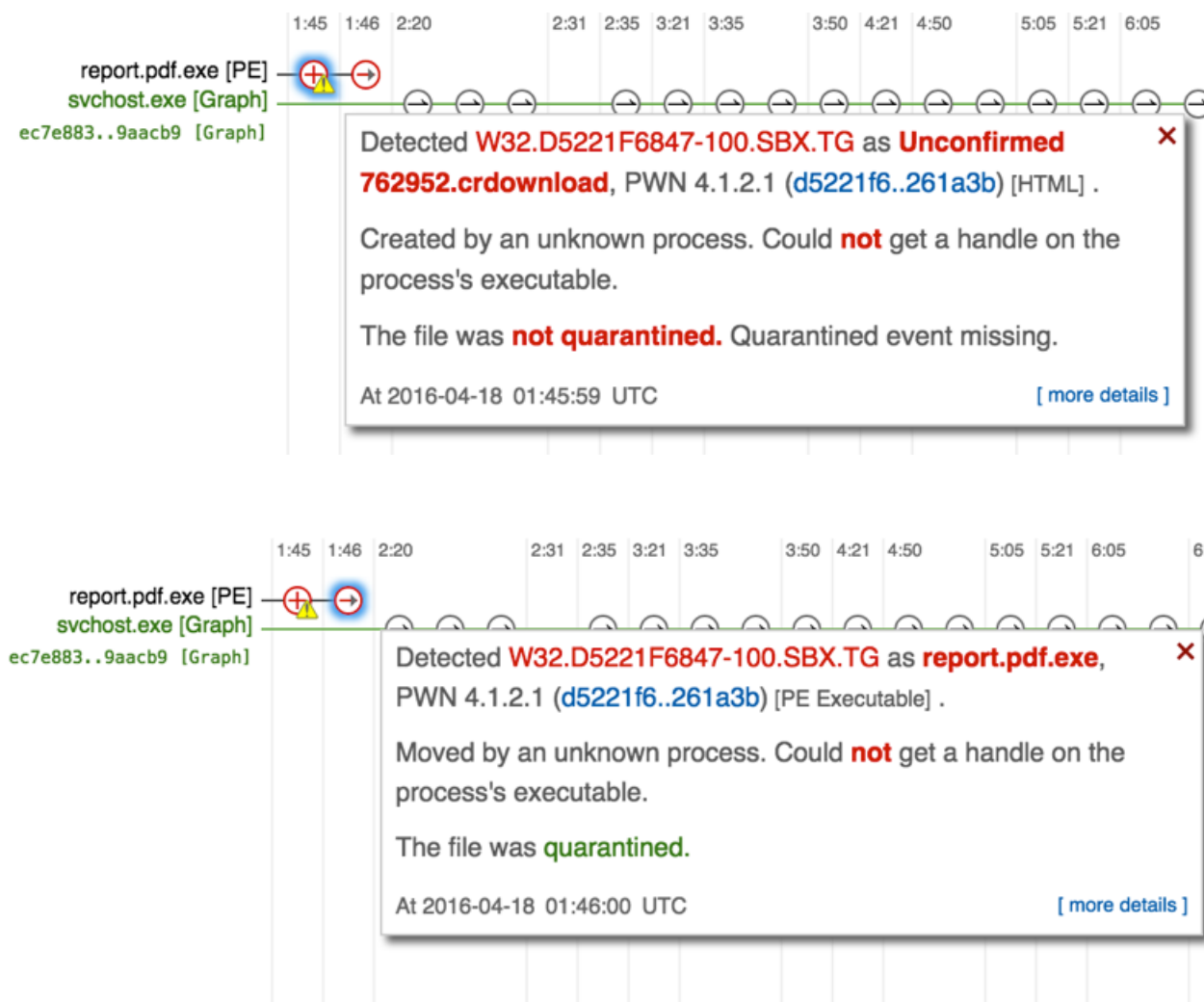
A number of suspicious domains are also being connected to by this executable:



Later on we see a detection and then a quarantine event occur for this file:

## Automatic Analysis of Low Prevalence Files, Resulting in a Retrospective Quarantine

### 3.0 Detection and Remediation



The first event is for the Google Chrome temporary download file, and the second event is for the file move operation to the original file name.

This is great to know, but how did it occur?

## 3.2 Retrospective Detection and Remediation Due to Low Prevalence

We look at the events for the computer by clicking the **Events** link on the **Computers** page:

Demo_Low_Prev_Retro in group Default Group	
Hostname	Demo_Low_Prev_Retro
Operating System	Windows XP, SP 3.0
Connector Version	3.1.4.9373
Install Date	2016-04-18 23:30:31 UTC
Connector GUID	d1fe8b4f-5002-42c3-9478-6d7f0c30960d
Events Launch Device Trajectory View Changes	

We see that a number of events have occurred on this computer. The ones we'd like to focus on are those that led us to the detection of this file.

The first event is a remote file fetch, which occurred due to the file having low prevalence throughout the organization. The file was fetched in order to be analyzed by Threat Grid to determine if it is in fact malicious:

Demo_Low_Prev_Retro requested a file			File Fetch Success
--------------------------------------	--	--	--------------------

The second is a resulting detection event due to the file receiving a Threat Score of **100** within Threat Grid:

Demo_Low_Prev_Retro detected report.pdf.exe as a malicious file during Low Prevalence Executable Analysis
---

To view the complete analysis report, click on the event to expand it, and then click the **Analysis results** link for this file:

Demo_Low_Prev_Retro detected report.pdf.exe as a malicious file during Low Pr...				Low Prevalence Exe...
File Detection	Fingerprint (SHA-256):	d5221f68...f5261a3b		
Connector Info	Filename:	report.pdf.exe		
Comments	Parent:	No parent SHA/Filename available.		
		Analysis results (1)		Add to V




Click the **Report** button on the resulting page to see the fully rendered HTML version of the report:

## File Analysis

For d5221f68...f5261a3b

[Download Sample](#) [Analysis Video](#) [Download PCAP](#) [38 Artifacts](#)



Metadata Behavioral Indicators Network Activity Processes Artifacts Registry Activity File Activity

### Analysis Report

<b>ID</b>	3a772d35918d53fc5497ce0c6e e719e9	<b>Filename</b>	report.pdf.exe
<b>OS</b>	2600.xpsp.080413-2111	<b>Magic Ty</b>	PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS
<b>Starte</b>	4/25/16 17:21:38	<b>pe</b>	Windows, UPX compressed
<b>d</b>		<b>Analyzed</b>	exe
<b>Ended</b>	4/25/16 17:29:32	<b>As</b>	
<b>Durati</b>	0:07:54	<b>SHA256</b>	d5221f6847978682234cb8ebfa951cb56b1323658679a820b168bbe1f 5261a3b
<b>on</b>		<b>SHA1</b>	5058b16a86beee96927371210b9a9f682976a50a
<b>Sandb</b>	test-work-03 (pilot-d)	<b>MD5</b>	48a0bf05b9706a00d2a0ff6260412f11
<b>ox</b>			

**Warnings**

- Executable Failed Integrity Check

### Behavioral Indicators

Process Enabled Autorun through the Creation of autorun.inf	Severity: 100	Confidence: 100
Outbound HTTP GET Request	Severity: 75	Confidence: 75
Process Modified File in a User Directory	Severity: 70	Confidence: 80
Potential Code Injection Detected	Severity: 50	Confidence: 50
PE Has Sections Marked Executable and Writable	Severity: 40	Confidence: 60
PE Contains TLS Callback Entries	Severity: 40	Confidence: 60

We can see in the above screenshot that the file received a Threat Score of **100** due to the file (launched and executing as a process) creating an *autorun.inf* file, which is commonly used by malware to spread infections throughout a network using file shares.

Once Threat Grid determined that the file was malicious, our Cloud Recall technology retrospectively detected and remediated the malicious file. This resulted in a Cloud Recall Quarantine Attempt event:

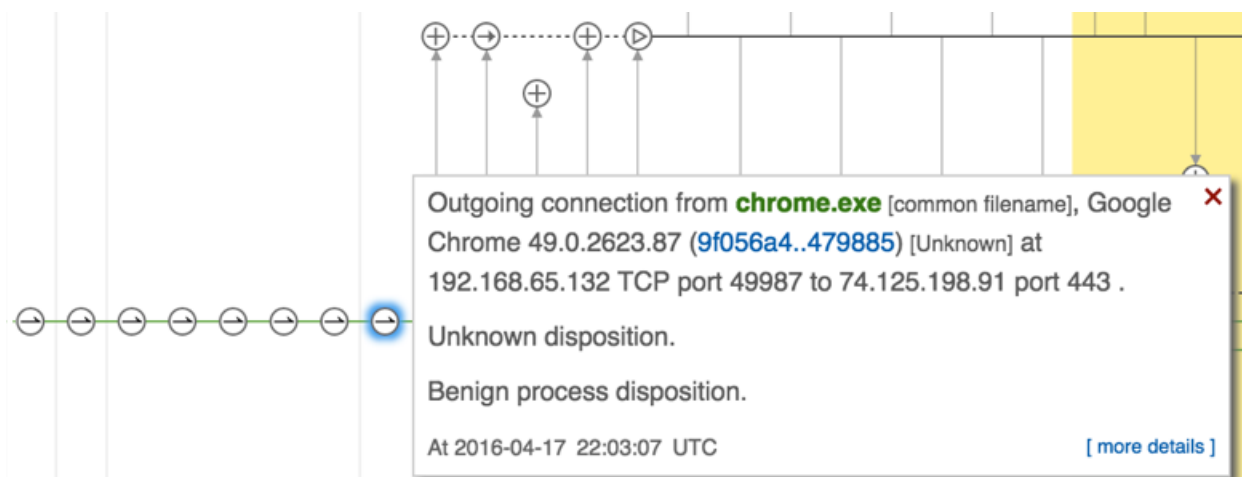
## Automatic Analysis of Low Prevalence Files, Resulting in a Retrospective Quarantine

### 3.0 Detection and Remediation

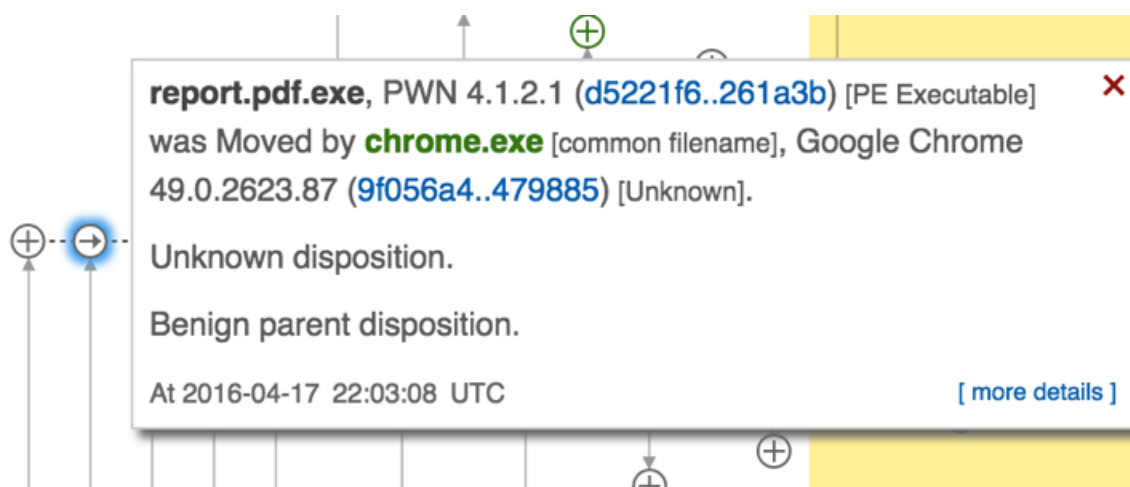
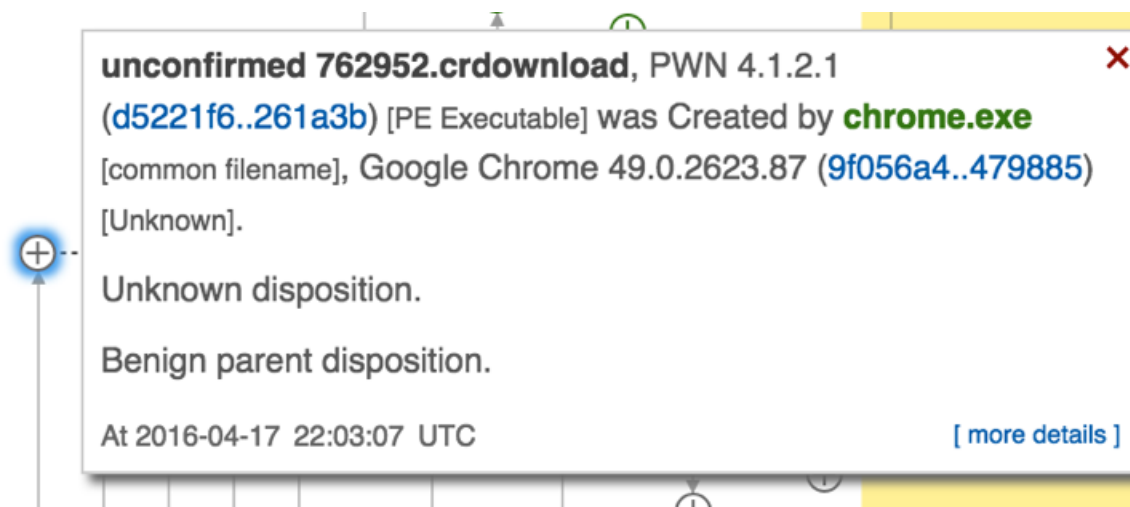


## 3.2 How it Started

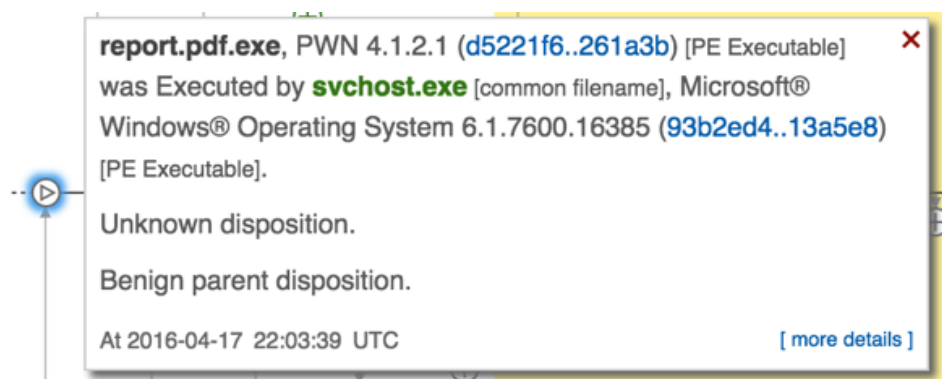
Now that we've discovered how this file was detected and remediated, we can use Device Trajectory to trace the attack back to its origins. Scrolling back to the earliest timestamp, we see a number of connections being made from Google Chrome (chrome.exe) on port 443:



We can see chrome.exe creating two separate files: the files that were detected in the Cloud Recall events:



Finally, once the file was downloaded we could see its execution:



Based on the information available we can deduce that one of the IP addresses that Google Chrome connected to was the source of the malicious executable, and that this file was executed once it was downloaded.

## 4.0 Summary

We determined through device trajectory in FireAMP that the delivery method of this attack was the execution of a masqueraded executable downloaded from Google Chrome.

We next determined that a previously unknown sample was uploaded to be analyzed using Remote File Fetch due to its low prevalence throughout the organization and was later convicted, resulting in retrospective remediation by FireAMP's Cloud Recall technology.

Through the file's analysis in Threat Grid we determined that the file had malicious intent and potential spreading capabilities.