CryptoWall

September, 2015

CONTENTS

PREFACE

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL 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.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED 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.

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.

1.0 Introduction

CryptoWall 3.0 is a type of malware known as ransomware. Upon infecting the host, the malware scans files on the infected host and encrypts them with a 2048 byte RSA public key encryption, making the files inaccessible. Once the files are encrypted, CryptoWall opens Notepad and displays instructions on how to decrypt the files. The instructions demand that you pay a ransom to purchase the decryption program. The ransom is initially $500; after a week the ransom increases to $1,000 and must be paid in Bitcoin.

The following scenario describes an encounter with a CryptoWall infection in the wild, in which FireAMP is first used to detect anomalous activity on the endpoint, and is then used to trace the attack back to its initial infection vector.

IMPORTANT!
In the following scenario the policy for the FireAMP 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 FireAMP. Audit mode will cause FireAMP to **not** terminate malicious processes or quarantine malicious files.

2.0 The Attack

The attack is a simple yet effective use of the macro functionality within a Microsoft Word document. It is delivered through a spear-phishing e-mail attack that downloads and executes a payload. The payload performs a number of dropping tasks that eventually lead to a CryptoWall infection.

3.0 Detection and Remediation

When you login to the FireAMP Console, the first page you see 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 that allows you to perform triage to determine which computers are in most need of immediate attention.

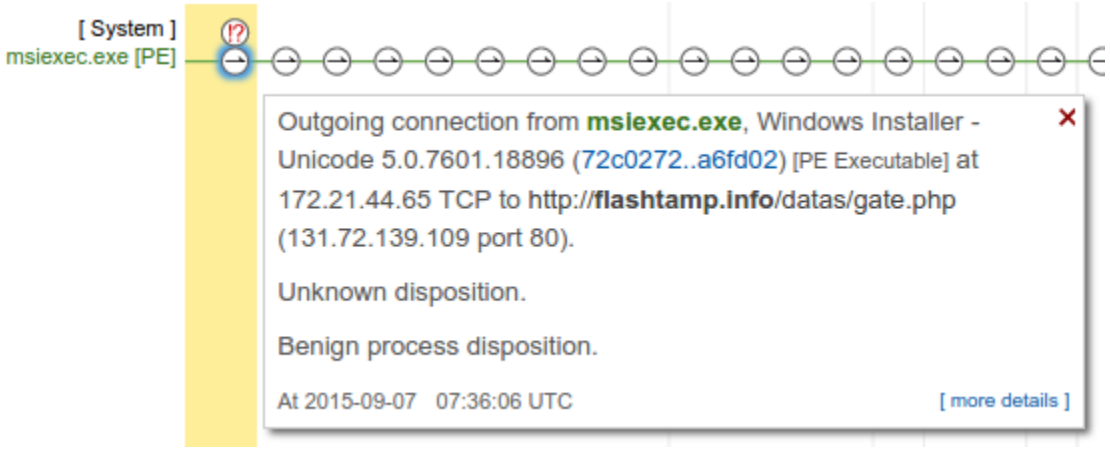
The Indications of Compromise on the Dashboard Overview helps with triage by listing computers with multiple events or separate events that correlate with certain types of infections. In our scenario we see that the top computers with indications of compromise have experienced Generic IOC detections.

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.

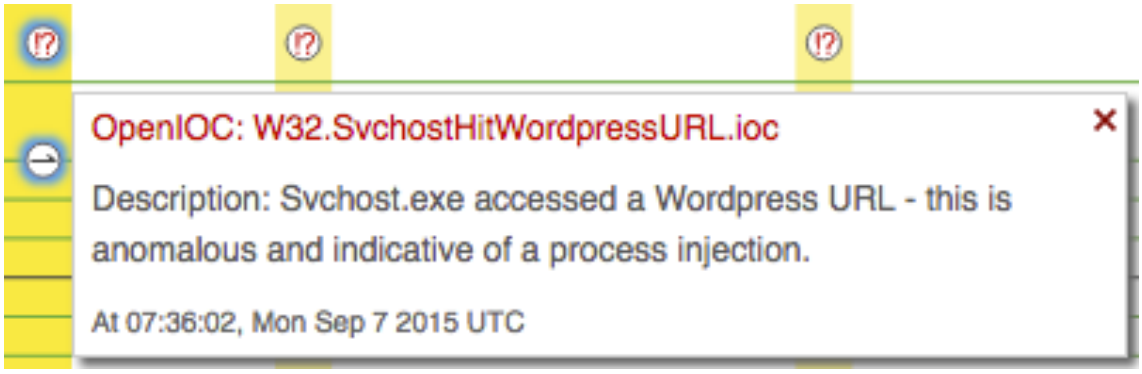
3.1 Tracing Back

Upon opening the Device Trajectory for one of the Generic IOC Detections we see an Indication of Compromise due to a suspicious URL being accessed (*gate.php*):

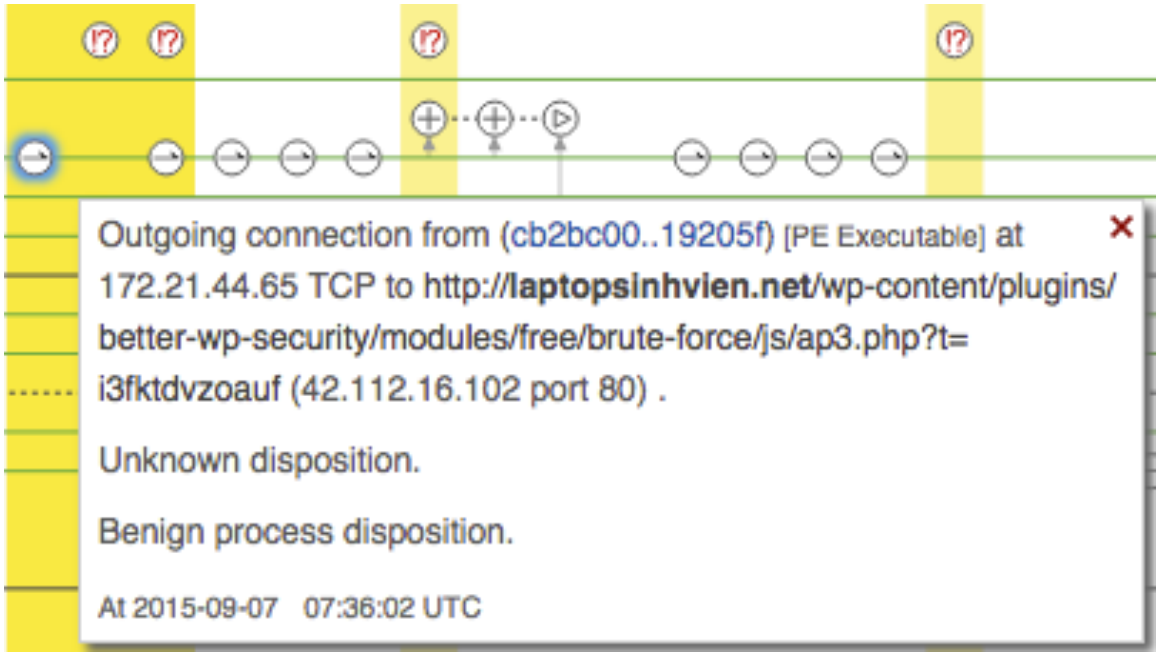


The event is coming from msiexec.exe – normally used to install MSI installers. In this case, it is making suspicious and regular outgoing connections, hitting shady websites. We can conclude is it infected and likely running malicious code.

In addition, we can see another Indicator of Compromise associated with svchost.exe that is behaving suspiciously. It is accessing a number of WordPress websites:

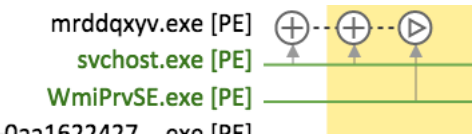


The Research & Efficacy Team has found a URL pattern that is associated with CryptoWall campaigns, and they have released an Indicator of Compromise to detect it.



The associated URL connection being made by svchost.exe that triggered the aforementioned IOCs is highlighted within the Device Trajectory:

We later see that svchost.exe has also created and executed a suspicious binary through WmiPrvSE.exe – a recent malware trick to hide the parent of an execution:

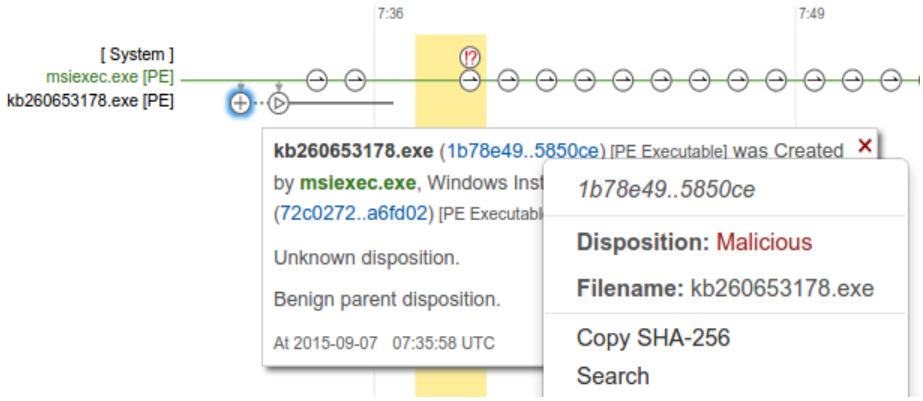


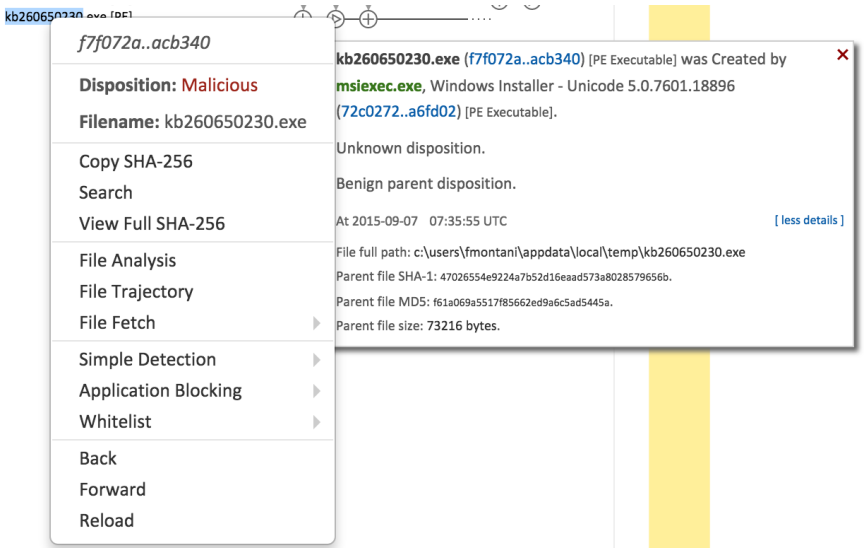
Clearly, svchost.exe is also infected. The system appears to have been compromised with at least two infected processes: msiexec.exe and svchost.exe.

Returning to our compromise, we need to find how all of this started.

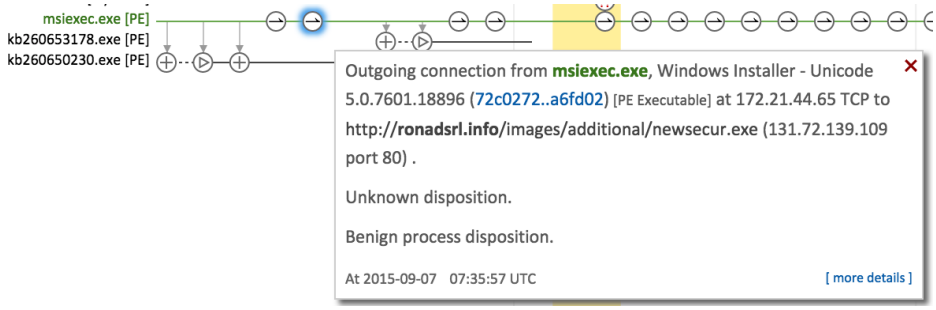
3.2 How It Started

Tracing back, we see that msiexec.exe created and executed two malicious files that have random but similar names (*kb[numeric].exe*):

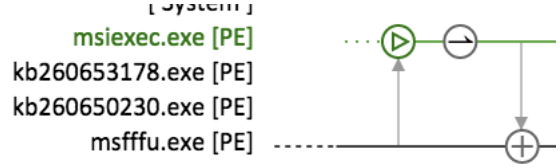




The source of the kb binaries can be seen from surrounding network connections, for example:

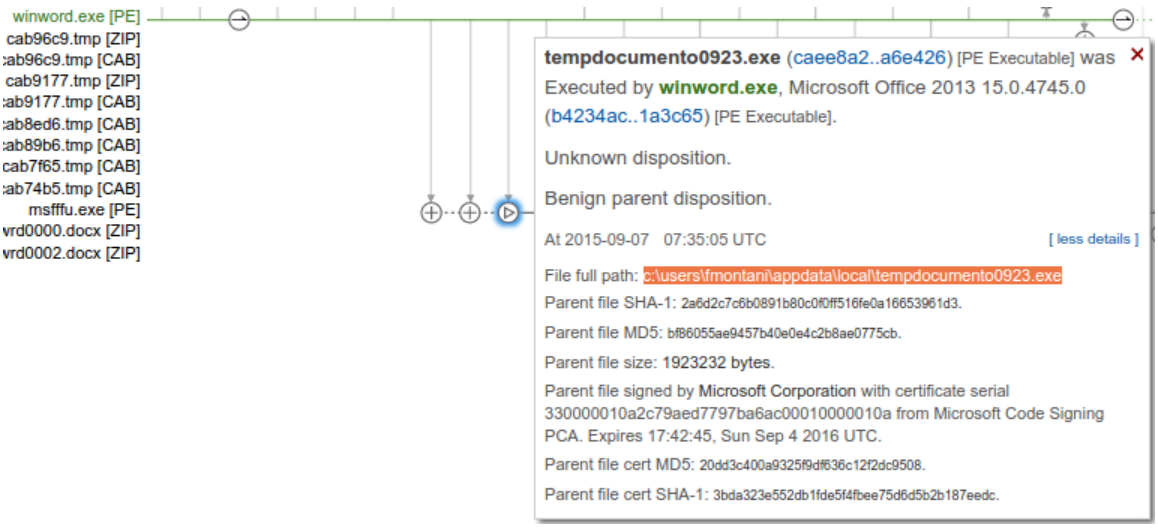


Tracing further back, we see that msiexec.exe was launched by a file with a random looking name (msfffu.exe):

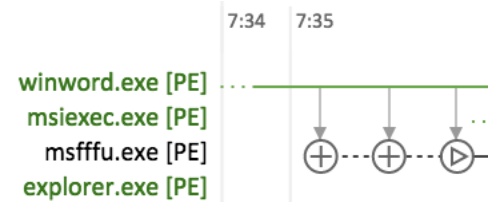


This is unusual, anomalous, and likely due to process hollowing – it confirms our suspicions that msiexec.exe has been actually running malicious code.

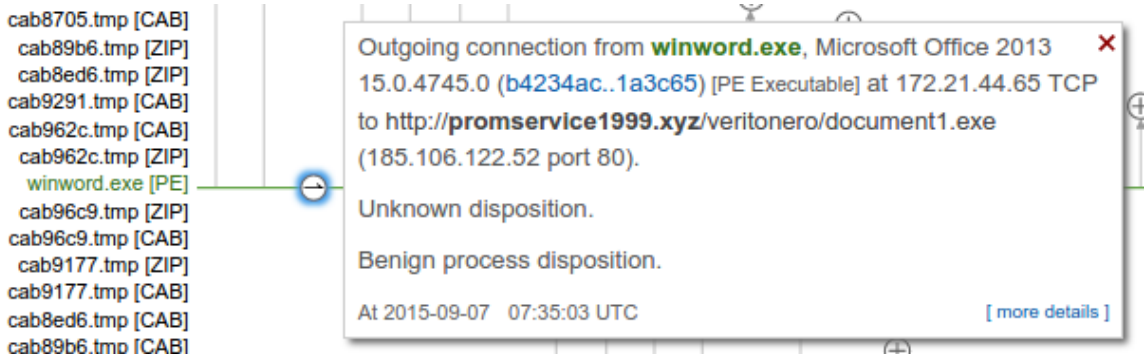
We now have the next link to trace back even further:



We can see that msfffu.exe was created and executed by winword.exe:

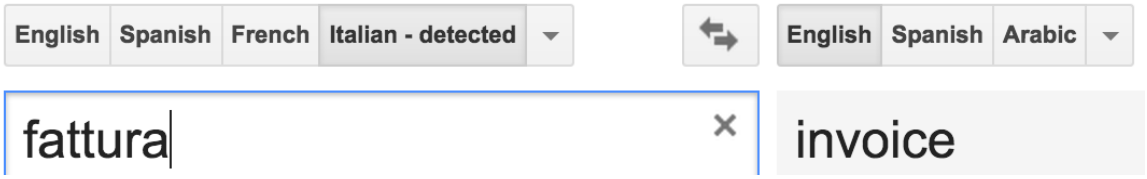


Looking back further, we can also see that msfffu.exe was downloaded from a suspicious domain (*promservice1999.xyz*):

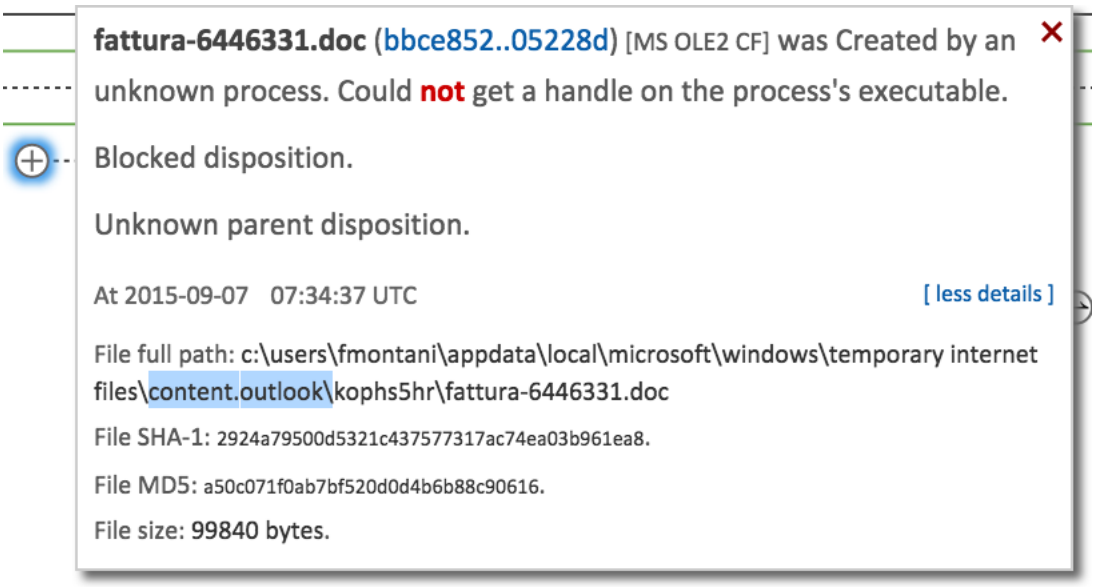


Clearly, Microsoft Word has been compromised here – it was either exploited, or it was used to open a malicious macro-laced document. Going back to the start of the winword.exe process, we see that a document was created, and winword was likely launched to open it:

This document is named *fattura-6446331.doc*, which translates to “invoice-6446331.doc” in Italian. Similarly named documents have been seen in several spear-phishing attacks, so it confirms our suspicions that this document was the original attack vector.



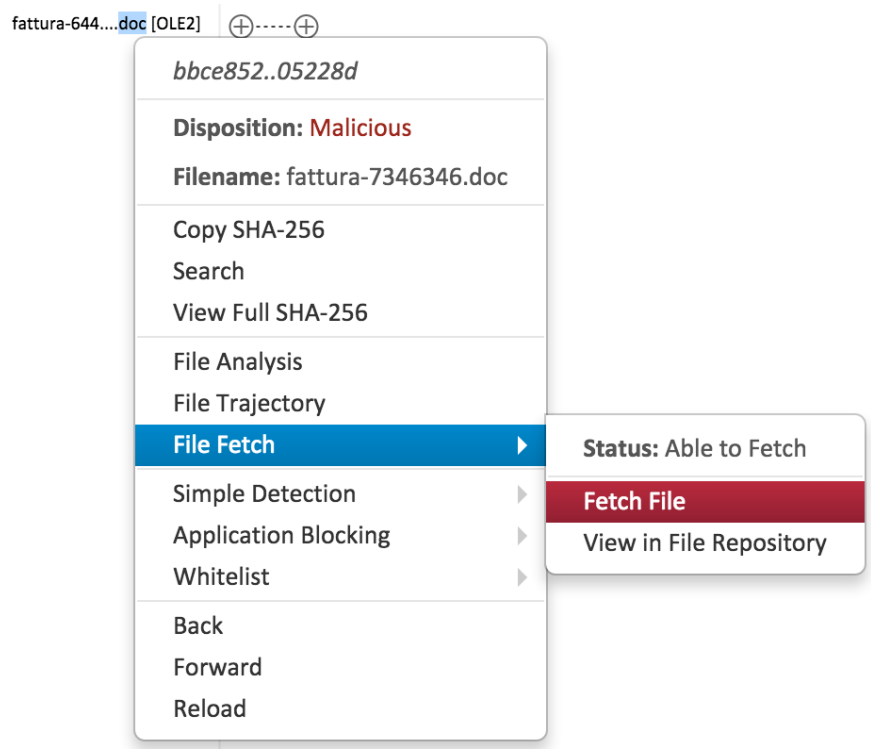
While the connector could not get a handle on the parent process, we know it was Microsoft Outlook based on the path of the document:



At this stage, we have the following information:

a. The host is breached.
b. The breach started with a Word document.
c. The Word document was received in an enticing email.

This is a typical attack scenario. What we are missing are details of what kind of malware has infected the host. To see the full range of behaviors, the file can be fetched and submitted to Threat Grid for analysis:



It becomes evident very quickly that this sample is malicious once it is submitted and analyzed by Threat Grid: The Threat Score is 100.

Behavioral Indicators

Threat Score: 100

Cryptowall 3.0 Detected	Severity: 100 Confidence: 100
Document Created an Executable File	Severity: 100 Confidence: 100
A Document File Established Network Communications	Severity: 100 Confidence: 90
BCDEdit Used to Modify Boot Options	Severity: 80 Confidence: 100
Shadow Copy Deletion Detected	Severity: 75 Confidence: 100
Process Attempted to Access the FireFox Password Manager Local Database	Severity: 95 Confidence: 75
Process Modified an Executable File	Severity: 60 Confidence: 100
Outbound HTTP GET Request	Severity: 75 Confidence: 75
Process Modified File in a User Directory	Severity: 70 Confidence: 80
Downloaded PE Executable	Severity: 60 Confidence: 90
Process Disabled Internet Explorer Proxy	Severity: 70 Confidence: 70
Process Modified Shell Program Autorun Registry Key Value	Severity: 80 Confidence: 60
Process Modified Autorun Registry Key Value	Severity: 80 Confidence: 60
Very Large Registry Data	Severity: 50 Confidence: 80
Process Created a File in the Windows Start Menu Folder	Severity: 80 Confidence: 50
Process Disables Explorer's Display of Hidden Files	Severity: 50 Confidence: 60
WMIC Used to Launch a Process	Severity: 50 Confidence: 60
Process Disables User Account Control (UAC) Settings	Severity: 50 Confidence: 60
File Downloaded to Disk	Severity: 30 Confidence: 90
File Name of Executable on Disk Does Not Match Original File Name	Severity: 40 Confidence: 60
DNS Query Returned Non-Existent Domain	Severity: 25 Confidence: 75
Possible Double Flux Nameserver Detected [Beta]	Severity: 35 Confidence: 50
Executable with Encrypted Sections	Severity: 30 Confidence: 30
Base64-Encoded Public Key Stored In Registry	Severity: 10 Confidence: 75
DNS Response Contains Low Time to Live (TTL) Value	Severity: 35 Confidence: 20
Outbound HTTP POST Communications	Severity: 25 Confidence: 25
Outbound Communications to Nginx Web Server	Severity: 25 Confidence: 25
Executable Imported the IsDebuggerPresent Symbol	Severity: 20 Confidence: 20

We see that at some point the sample has triggered a behavioral indicator, indicating it is CryptoWall 3.0. Cisco's Talos has published an excellent write up on CryptoWall 3.0 that can be reviewed for further details on the workings of CryptoWall 3.0:

http://blogs.cisco.com/security/talos/cryptowall-3-0

In the sample we highlight several important aspects that help determine the sample to be malicious. It's important however to realize this is a view of several pieces of the puzzle combined in a full execution. We have the document, and the components the document drops.

We highlight the two main components of the document, which, by itself would cause this sample to be deemed malicious:
1) Creating an executable
2) Establishing network communications

Next we find evidence consistent with that of ransomware: we generally find all variants of ransomware deleting shadow copies, which would otherwise allow for the recovery of encrypted content. In the event a sample is submitted which does not match the behavioral indicators specific to a family of malware, if this indicator is present the malware is likely a ransomware variant.

In addition to the deletion of shadow copies, we find a process making calls to the Windows utility bcdedit.exe, modifying the boot options on the host:

BCDEdit Used to Modify Boot Options

The "BCDEdit" command displays and modifies information about the boot options for Windows Vista and later Windows operating systems. It allows changing the boot order, potential booting options and enabling recovery or debug options. Malware authors may modify the boot process to hide from system checks, for early presence on the system and to persist through reboots.

Command Line	Process Name	Process ID
bcdedit /set {default} bootstatuspolicy ignoreallfailures	bcdedit.exe	1492 (bcdedit.exe)
bcdedit /set {default} recoveryenabled No	bcdedit.exe	1328 (bcdedit.exe)

In this case, it is setting the system to ignore failures, and disabling recovery mode on the workstation.

We also find that both MSIExec and Explorer have added persistence components:

Process Modified Autorun Registry Key Value — **Severity: 80 Confidence: 60**

Autorun registry keys can be used to load applications when Windows is started. Malware often uses these key locations to maintain persistence on the host. The values to examine are located in subkeys Run, RunOnce, RunServices, RunServicesOnce, RunOnceEx, or RunOnce\Setup. The key value will indicate where the program that will load on startup is located.

Categories persistence
Tags process, autorun, registry

Report Error

RegKey Data	RegKey Data Type	RegKey Value Name	RegKey Name	Process Name	Process ID
C:\Users\Administrator\AppData\Roaming \3aaa1fab.exes\\0	SZ	3aaa1fab	USER\S-1-5-21-3980782205-2872255537-1618903166-500 \SOFTWARE\MICROSOFT\WINDOWS \CURRENTVERSION\RUN	explorer.exe	1288 (explorer.exe)
C:\3aaa1fab\3aaa1fab.exes\\0	SZ	3aaa1fa	USER\S-1-5-21-3980782205-2872255537-1618903166-500 \SOFTWARE\MICROSOFT\WINDOWS \CURRENTVERSION\RUN	explorer.exe	1288 (explorer.exe)
C:\3aaa1fab\3aaa1fab.exes\\0	SZ	*aaa1fa	USER\S-1-5-21-3980782205-2872255537-1618903166-500 \SOFTWARE\MICROSOFT\WINDOWS \CURRENTVERSION\RUNONCE	explorer.exe	1288 (explorer.exe)
C:\Users\Administrator\AppData\Roaming \3aaa1fab.exes\\0	SZ	*aaa1fab	USER\S-1-5-21-3980782205-2872255537-1618903166-500 \SOFTWARE\MICROSOFT\WINDOWS \CURRENTVERSION\RUNONCE	explorer.exe	1288 (explorer.exe)

Network Traffic is presented in Threat Grid reports in the order it was seen in the packet capture. The first item we see is the downloading of document1.exe, which is the executable downloaded by the document that triggered our earlier indicator in AMP:

GET http://promservice1999.xyz:80/veritonero/document1.exe

Server IP: 185.106.122.52 **Server Port:** 80

Method	GET
URL	http://promservice1999.xyz:80/veritonero/document1.exe
Request	-
Timestamp	+89.426s
Actual Encoding	
Actual Content-type	application/x-empty; charset=binary

Header	Value
host	promservice1999.xyz
connection	Keep-Alive
accept	*/*
accept-encoding	gzip, deflate
user-agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET CLR 1.1.4322; .NET4.0C; .NET4.0E)

The document1.exe download and execution is followed by a request for what appears to be a certificate:

POST http://studiofinaxtec.info:80/comodo/ser.php

Server IP: 131.72.136.91 **Server Port:** 80

Method	POST
URL	http://studiofinaxtec.info:80/comodo/ser.php
Request	-
Timestamp	+95.709s
Actual Encoding	windows-1252
Actual Content-type	application/octet-stream; charset=binary

Header	Value
cache-control	no-cache
pragma	no-cache
content-length	70
user-agent	Mozilla/4.0
host	studiofinaxtec.info
content-type	application/octet-stream
connection	close

Then we see a request for an additional binary called newsecur.exe:

GET http://ronadsrl.info:80/images/additional/newsecur.exe

Server IP: 131.72.139.109 **Server Port:** 80

Method	GET
URL	http://ronadsrl.info:80/images/additional/newsecur.exe
Request	-
Timestamp	+98.076s
Actual Encoding	
Actual Content-type	application/x-empty; charset=binary

Header	Value
cache-control	no-cache
host	ronadsrl.info
connection	close
pragma	no-cache
user-agent	Mozilla/4.0

This is the original domain that triggered the AMP Event:

POST http://flashtamp.info:80/datas/gate.php

Server IP: 131.72.139.109 **Server Port:** 80

Method	POST
URL	http://flashtamp.info:80/datas/gate.php
Request	-
Timestamp	+106.005s
Actual Encoding	windows-1252
Actual Content-type	application/octet-stream; charset=binary

Header	Value
content-length	2468
accept-encoding	identity, *;q=0
user-agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET CLR 1.1.4322; .NET4.0C; .NET4.0E)
host	flashtamp.info
accept-language	en-US
accept	*/*
content-type	application/octet-stream
connection	close
content-encoding	binary

We don't see all the same domains in the sample run in Threat Grid that we did from the AMP Device Trajectory. This is due largely to the behavior of CryptoWall 3.0 rotating through a list of domains that it contacts for its Command & Control. This observation is present in the last 3 requests made by the workstation where the URL Path indicates it is communicating with a PHP file on a website that has WordPress installed, and very closely matches that of the observed URL Pattern in the AMP Compromise:

POST http://linecellardemo.net:80/wp-content/plugins/wp-db-backup-made/ap4.php?v=wcm1xj9904z3a4

Server IP: 23.229.194.224 **Server Port:** 80 **Resp. Content:** t

Method	POST
URL	http://linecellardemo.net:80/wp-content/plugins/wp-db-backup-made/ap4.php?v=wcm1xj9904z3a4
Request	-
Timestamp	+110.247s
Actual Encoding	ascii
Actual Content-type	text/plain; charset=us-ascii
Response Timestamp	
Actual Encodin	
Actual Content	
Artifact ID	

Header	Value
host	linecellardemo.net
content-length	136
accept	*/*
cache-control	no-cache
connection	Close
content-type	application/x-www-form-urlencoded
user-agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET CLR 1.1.4322; .NET4.0C; .NET4.0E)

Header
date
transfer-encodin
x-powered-by
connection
content-type
server

Pivoting in Threat Grid gives us greater visibility into the global data set. We can quickly identify additional samples, and from them gather additional resources in our efforts to remediate and block attacks.

The first domain does not yield much information, in fact the only related sample listed is that of the sample we were just analyzing.

Pivoting on the domain responsible for the certificate request we find a large number of related samples:

https://panacea.threatgrid.com/domains/studiofinaxtec.info

Samples related to studiofinaxtec.info

Showing items 1 - 50 << >>

Sample	Sha256	Indicator Summary	Relation	Time
53d42ce0afd6c163320bda2b496408db	9ca87de01c5e2e8a...	9 / 80 / 100	dns-lookup	9/17/15 18:06:44
5ad5eb172fe4bb7651cd7f2a32764714	8a2ee2c44408b90e...	9 / 80 / 100	dns-lookup	9/17/15 17:41:43
5ad5eb172fe4bb7651cd7f2a32764714	8a2ee2c44408b90e...	9 / 80 / 100	http-requests	9/17/15 17:41:43
542f27c50e26176515fa2bf1ffa2d2b1	4e3337752a549238...	9 / 80 / 100	dns-lookup	9/17/15 17:13:24
542f27c50e26176515fa2bf1ffa2d2b1	4e3337752a549238...	9 / 80 / 100	http-requests	9/17/15 17:13:24
c404df279df38b07a5896d71c9befbcd	9c12a1a5017db86d...	9 / 80 / 100	dns-lookup	9/17/15 14:53:59
c404df279df38b07a5896d71c9befbcd	9c12a1a5017db86d...	9 / 80 / 100	http-requests	9/17/15 14:53:59
33952e14014df27d4a5b0abd3a3492ff	e73d0a70b2cb284f...	14 / 80 / 100	dns-lookup	9/17/15 14:34:12
33952e14014df27d4a5b0abd3a3492ff	e73d0a70b2cb284f...	14 / 80 / 100	http-requests	9/17/15 14:34:12
85b0fe4ca5df710ab893ea04ec2db42a	5df430c1661e5d90...	11 / 80 / 100	dns-lookup	9/17/15 13:33:12
85b0fe4ca5df710ab893ea04ec2db42a	5df430c1661e5d90...	11 / 80 / 100	http-requests	9/17/15 13:33:12
b4b0dbe38e85bac99229e5d8f73ad3a1	14ff6262bdc87e37...	11 / 80 / 100	dns-lookup	9/17/15 11:48:39
b4b0dbe38e85bac99229e5d8f73ad3a1	14ff6262bdc87e37...	11 / 80 / 100	http-requests	9/17/15 11:48:39
49412c7dda774c4159cd73f82e344451	4a4250029e9805f5...	8 / 80 / 100	dns-lookup	9/17/15 11:09:48
0a0e56df0a728968794bd52972968bee	d3c3c95548abef52...	11 / 80 / 100	dns-lookup	9/17/15 11:13:33
0a0e56df0a728968794bd52972968bee	d3c3c95548abef52...	11 / 80 / 100	http-requests	9/17/15 11:13:33
1036245921	e75f91fe1ff08afd...	14 / 90 / 100	dns-lookup	9/17/15 09:50:32
1036245921	e75f91fe1ff08afd...	14 / 90 / 100	http-requests	9/17/15 09:50:32
1027869655	87a20785deaa0eb8...	13 / 90 / 100	dns-lookup	9/17/15 09:40:50
1027869655	87a20785deaa0eb8...	13 / 90 / 100	http-requests	9/17/15 09:40:50
33b6e12ddbf72993fc1a8fb65a3d4bcc	9b4cd1cc762536c9...	10 / 80 / 100	dns-lookup	9/17/15 04:12:05
33b6e12ddbf72993fc1a8fb65a3d4bcc	9b4cd1cc762536c9...	10 / 80 / 100	http-requests	9/17/15 04:12:05
cb842edb089ee8acbf71ea6c01db5c5a	e0e302f88d7e28f1...	11 / 80 / 100	dns-lookup	9/17/15 03:03:06
cb842edb089ee8acbf71ea6c01db5c5a	e0e302f88d7e28f1...	11 / 80 / 100	http-requests	9/17/15 03:03:06
16b5b47973fa39ed8135180fc0d56653	eb392b331e68b0a0...	10 / 80 / 100	dns-lookup	9/17/15 02:46:35
16b5b47973fa39ed8135180fc0d56653	eb392b331e68b0a0...	10 / 80 / 100	http-requests	9/17/15 02:46:35
d30ec1e09cdc53d663fbe16cda3a29fc2837c8ca797e9fecf1c6a54e92f50c64	d30ec1e09cdc53d6...	13 / 90 / 100	dns-lookup	9/17/15 01:20:57
d30ec1e09cdc53d663fbe16cda3a29fc2837c8ca797e9fecf1c6a54e92f50c64	d30ec1e09cdc53d6...	13 / 90 / 100	http-requests	9/17/15 01:20:57
272cdb30e7de0ac85e467a6ab7b4bfa2	cb39d1853bb394b5...	10 / 80 / 100	dns-lookup	9/17/15 00:53:18
272cdb30e7de0ac85e467a6ab7b4bfa2	cb39d1853bb394b5...	10 / 80 / 100	http-requests	9/17/15 00:53:18

We also find the original domain responsible for the gate.php generic indicator in AMP containing quite a few related samples:

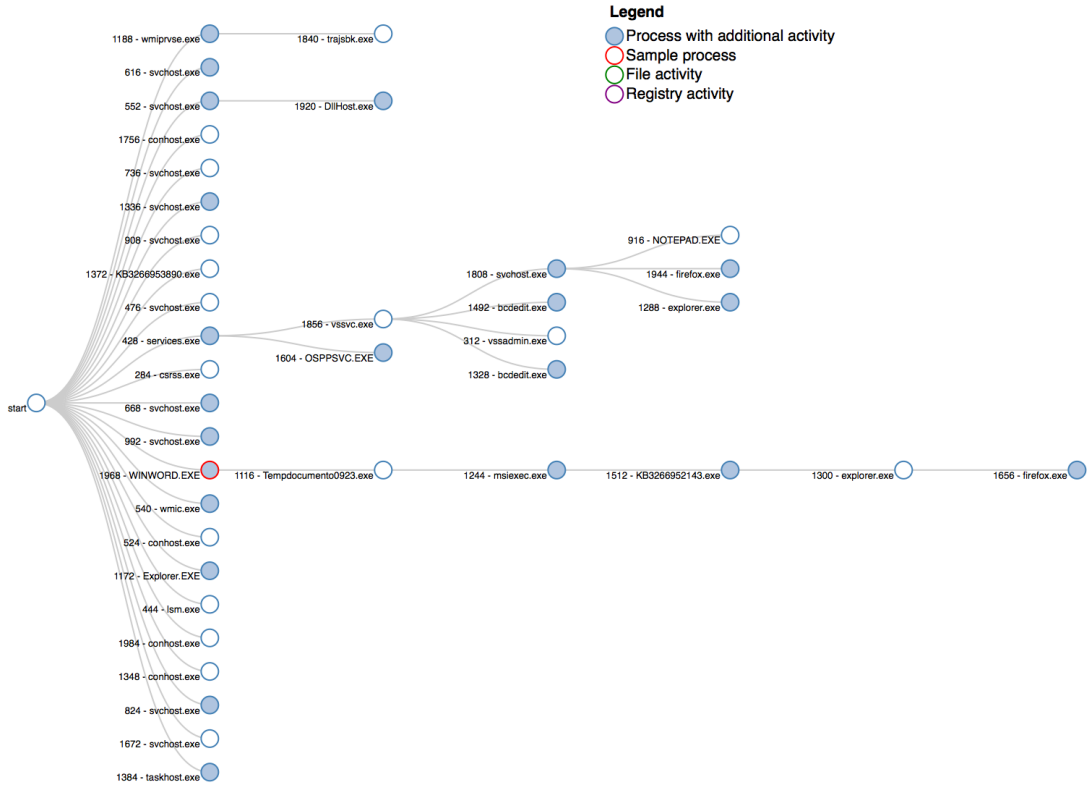
Samples related to flashtamp.info				
Sample	**Sha256**	**Indicator Summary**	**Relation**	**Time**
1023968699	6381a4c6c8a3d38d...	19 / 95 / 100	dns-lookup	9/16/15 13:52:21
1023968699	6381a4c6c8a3d38d...	19 / 95 / 100	http-requests	9/16/15 13:52:21
d180dd823c64cd8f2403f0b383de9cdc	e344beae21cdcb01...	20 / 95 / 100	dns-lookup	9/16/15 13:38:39
d180dd823c64cd8f2403f0b383de9cdc	e344beae21cdcb01...	20 / 95 / 100	http-requests	9/16/15 13:38:39
1574968ce67fd3591b4c222da1265929	55cbf03e4d8d77a5...	20 / 95 / 100	dns-lookup	9/16/15 12:07:45
1574968ce67fd3591b4c222da1265929	55cbf03e4d8d77a5...	20 / 95 / 100	http-requests	9/16/15 12:07:45
bbce85205a5f450ae33b80b648c7a56e9da932915185663350d0c39d0e05228d.doc	bbce85205a5f450a...	28 / 100 / 100	dns-lookup	9/9/15 17:35:39
bbce85205a5f450ae33b80b648c7a56e9da932915185663350d0c39d0e05228d.doc	bbce85205a5f450a...	28 / 100 / 100	http-requests	9/9/15 17:35:39
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	http-requests	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55
69489d774109d6f149ec504b772795f7	605535524cdfe693...	32 / 100 / 100	dns-lookup	8/24/15 14:31:55

Each of the related samples to flashtamp.info has a threat score of about 95, meaning the disposition of the sample will automatically be changed in AMP to Malicious.

The process listing provides a lot of information, but when large quantities of processes are present it can sometimes be difficult to decipher what is going on. An often-overlooked component of the Threat Grid Reports is the process graph, which displays this information visually:

https://panacea.threatgrid.com/samples/27c70f9894ba84d0f5ccde442e0a70bd/viz/processes.html

Process Tree for Sample 27c70f9894ba84d0f5ccde442e0a70bd



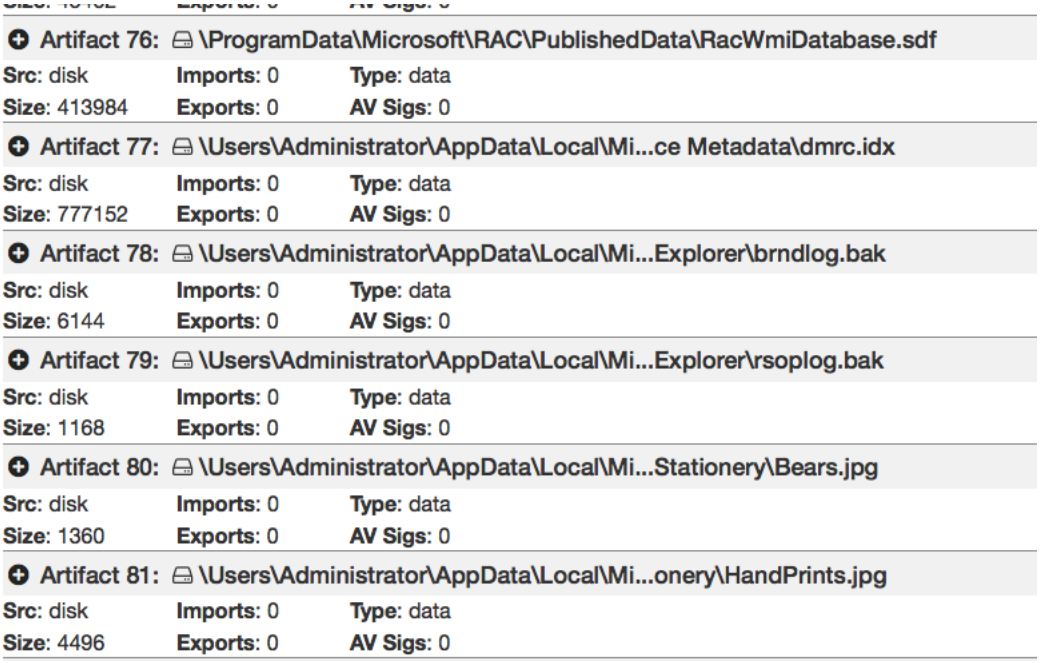
Our original sample's parent is highlighted in red, and we can see the execution path for this process launching additional processes.

We also see four additional paths of execution: CryptoWall 3.0 injects into the svchost.exe process through the services process. The code injected inside the svchost.exe process implements the main malware functionality.

After viewing the process graph we get a clear execution path, providing more details surrounding what happened.

Analyzing the Artifacts we find that svchost.exe has started modifying the files on disk. It starts by making a new entry in the root of every directory:

Then we find that svchost.exe starts modifying all the files on the system by encrypting them. This results in all files becoming just data and therefore can-not be opened in any application:



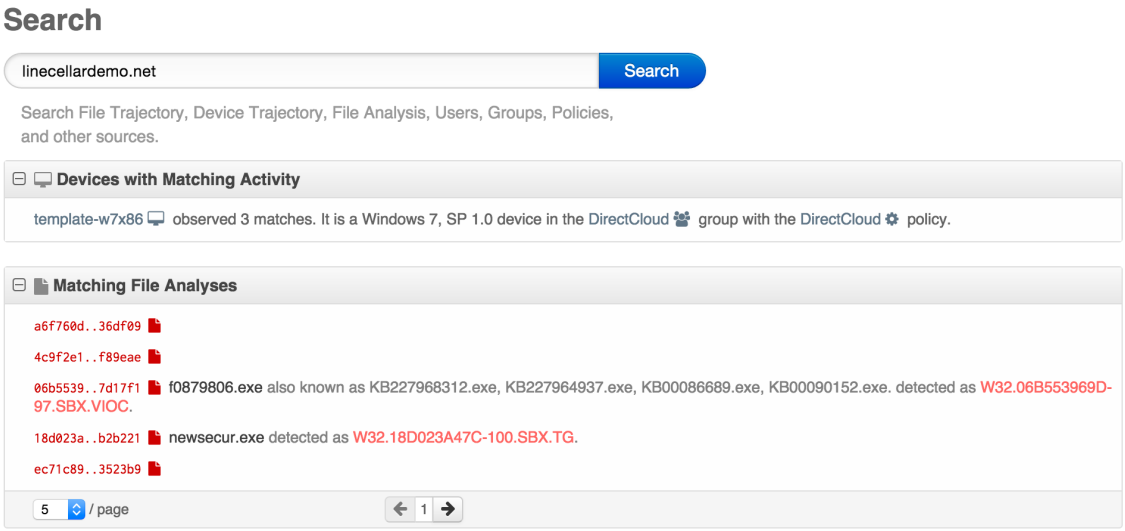
Threat Grid allows us to download any of the artifacts presented in the report through the artifact page or a simple Download Button on the right hand side of the artifact.

This allows us to view the contents of any of the files, or take them off line for further, manual analysis.

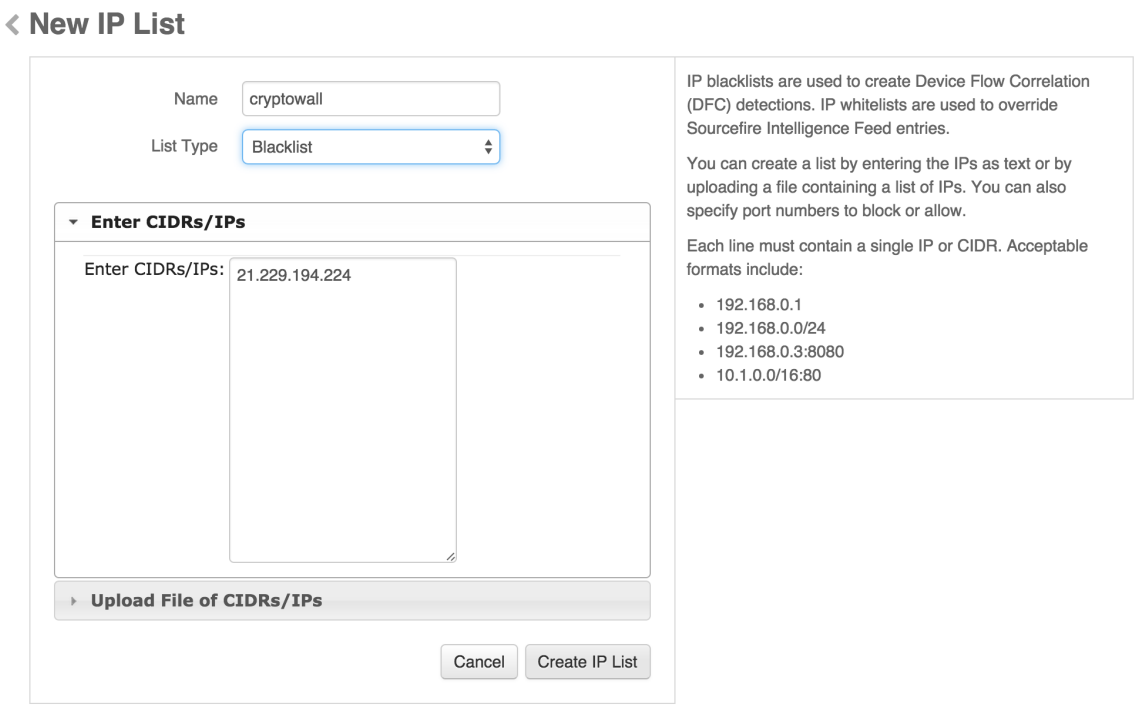
```
What happened to your files ?
All of your files were protected by a strong encryption with RSA-2048 using CryptoWall 3.0.
More information about the encryption keys using RSA-2048 can be found here: http://en.wikipedia.org/wiki/RSA_(cryptosystem)


What does this mean ?
This means that the structure and data within your files have been irrevocably changed, you will not be able to work with them, read them or
see them,
it is the same thing as losing them forever, but with our help, you can restore them.


How did this happen ?
Especially for you, on our server was generated the secret key pair RSA-2048 - public and private.
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of your files is only possible with the help of the private key and decrypt program, which is on our secret server.


What do I do ?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed.
If you really value your data, then we suggest you do not waste valuable time searching for other solutions because they do not exist.


For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:
1.http://6i3cb6owitcouepv.speralreaopio.com/px788m
2.http://6i3cb6owitcouepv.vremlreafpa.com/px788m
3.http://6i3cb6owitcouepv.wolfwallsreaetpay.com/px788m
4.http://6i3cb6owitcouepv.askhoreasption.com/px788m

If for some reasons the addresses are not available, follow these steps:
1.Download and install tor-browser: http://www.torproject.org/projects/torbrowser.html.en
2.After a successful installation, run the browser and wait for initialization.
3.Type in the address bar: 6i3cb6owitcouepv.onion/px788m
4.Follow the instructions on the site.


IMPORTANT INFORMATION:
Your personal page: http://6i3cb6owitcouepv.speralreaopio.com/px788m
Your personal page (using TOR): 6i3cb6owitcouepv.onion/px788m
Your personal identification number (if you open the site (or TOR 's) directly): px788m
```

3.3 Remediation

We can use the information derived from the Threat Grid report to identify further infections within our environment. As you can see, the domain has been indexed within a number of matching file analysis runs:



It appears as though this domain is being contacted by another machine within the environment. We can blacklist its associated IP address to be blocked by creating a new blacklist:



Using further information from the Threat Grid report we can create an OpenIOC that will identify infections using registry attributes. Threat Grid has identified the sample creating two autorun registry entries pointing to a randomly generated filename within the *C:\Documents and Settings\Administrator\Application Data* and *\b7f99a1* directories. This is fairly uncommon behavior, as legitimate applications should reside within the *Program Files* directory for startup purposes. The attributes can be defined as follows:

```
<IndicatorItem id="f6da22d3-5ce5-4d5d-b0cd-240924e26928" condition="contains" preserve-case="false" negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
  <Context document="RegistryItem" search="RegistryItem/Path" type="mir"/>
  <Content type="string">\Software\Microsoft\Windows\CurrentVersion\Run</Content>
</IndicatorItem>
<IndicatorItem id="545932aa-3684-46dd-bc0d-5222e480563b" condition="matches" preserve-case="false" negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
  <Context document="RegistryItem" search="RegistryItem/Text" type="mir"/>
  <Content type="string">.*(AppData|Application Data)(\\Roaming)?\\[a-f0-9]{7}\.exe</Content>
</IndicatorItem>


<IndicatorItem id="ffa643ea-0b7a-418d-9e22-85669a048539" condition="contains" preserve-case="false" negate="false" group-id="7b4bfbc1-a135-45a0-82d9-801ce0e10e81">
  <Context document="RegistryItem" search="RegistryItem/Path" type="mir"/>
  <Content type="string">\Software\Microsoft\Windows\CurrentVersion\Run</Content>
</IndicatorItem>
<IndicatorItem id="a35d27e0-dd71-4f33-839a-a2b162ee728d" condition="matches" preserve-case="false" negate="false" group-id="7b4bfbc1-a135-45a0-82d9-801ce0e10e81">
  <Context document="RegistryItem" search="RegistryItem/Text" type="mir"/>
  <Content type="string">[A-Z]:\\[a-f0-9]{7}\\[a-f0-9]{7}\.exe</Content>
</IndicatorItem>
```

Subsequent runs indicate that the file/folder name is 7 random hexadecimal characters in each of these locations as was defined within the regular expression above. This

alone could possibly result in false positives, as this doesn't necessarily imply that a CryptoWall infection resides on the system. To narrow our search space we can look for further information that is less likely to correspond to legitimate PE executables. Threat Grid has also identified that the executable within this directory has imported *IsDebuggerPresent,* which is typically used by malware to check whether it is being debugged. If we dropdown all of the executables involved we see the following:

Executable Imported the IsDebuggerPresent Symbol

The IsDebuggerPresent function can be used by a process to check if a debugger has been attached to it, or is currently active on the system. Malware authors often check for the presence of a debugger as this is an indication that the malware is being analysed. The Malware may not run, or it may function differently, if a debugger is present, to make it more difficult to reverse-engineer its behavior. This is not an indicator of malicious activity as often legitimate programs import this function.

Path	Artifact ID
KB3266953890.exe	1
\Documents and Settings\Administrator\Application Data\b7f99a1.exe	3
\Documents and Settings\Administrator\Start Menu\Programs\Startup\b7f99a1.exe	4
\b7f99a1\b7f99a1.exe	2

This can be added to our OpenIOC as follows:

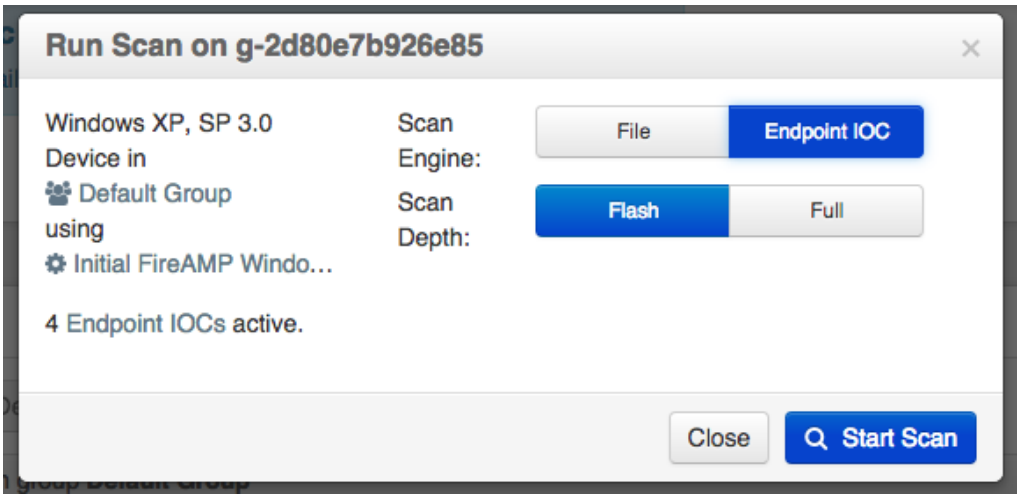
```
<IndicatorItem id="1f0390c2-5b92-4488-8abf-65d97016ed12" condition="contains" preserve-case="false" negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
  <Context document="FileItem" search="FileItem/PEInfo/ImportedModules/ImportedFunctions/string" type="mir"/>
  <Content type="string">IsDebuggerPresent</Content>
</IndicatorItem>
```

Both executables being pointed to by registry run keys are importing this. FireAMP provides two options for OpenIOC scanning. Flash Scans take much less time as they're only collecting a subset of attributes on the system. One of the recent enhancements to Flash Scan collections was to collect the executables residing at the path of registry run keys. We can use this to our advantage if we want to perform a quick check for this malware throughout the environment.

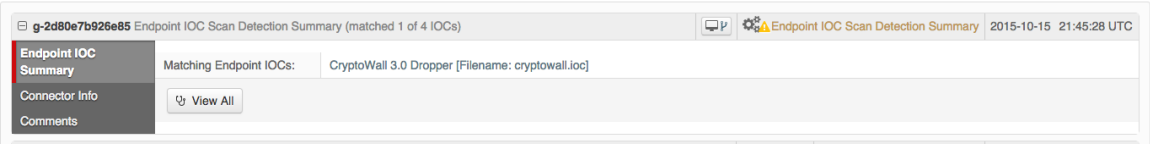
Alternatively, a Full Scan will collect all system attributes that are currently supported. This includes all filenames on the system. Within the Threat Grid report we see a ransom note being written that contains this filename: *HELP_DECRYPT.HTML.* We can add this to our OpenIOC within our OR statement that will be triggered from a Full Scan:

```
<Indicator id="cb9f5639-90b7-41ca-88e8-02398b8949a3" operator="OR">
  <IndicatorItem id="6b1d051a-0d56-43f0-93ae-5be4d8fb6b43" condition="contains" preserve-case="false" negate="false" group-id="e9b83582-6149-47cf-8fef-4382a59438db">
    <Context document="FileItem" search="FileItem/FileName" type="mir"/>
    <Content type="string">HELP_DECRYPT.HTML</Content>
  </IndicatorItem>
</Indicator>
```

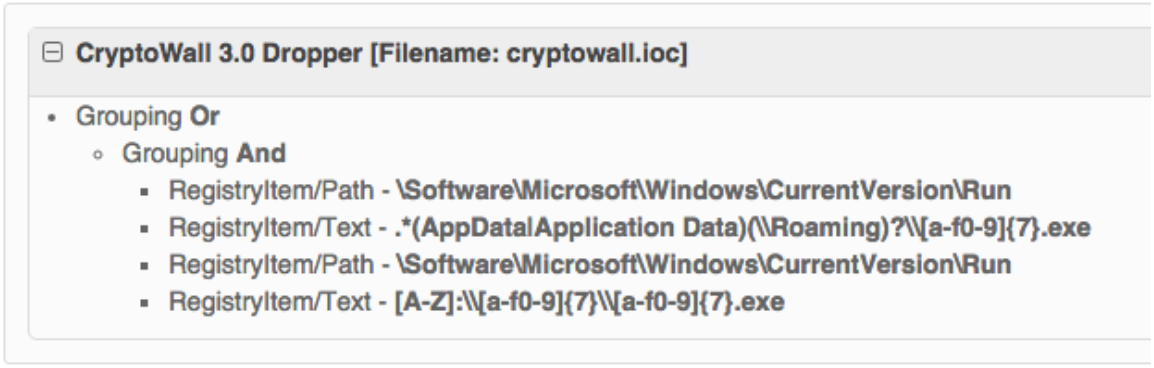
Now that our OpenIOC has come together we can initiate an Endpoint IOC flash scan:



To which we receive a hit within our environment:



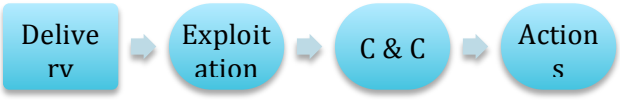
If we click on the “View All” button we can see what hit within this IOC:



Clicking on “View Source” provides us with highlights within the XML:

```
11      <Indicator id="297fe747-0f2f-4c54-9797-80a87505e73c" operator="OR">
12        <Indicator id="9e8f4b44-67e3-4dd6-9bd6-b0b90c8a4040" operator="AND">
13          <IndicatorItem id="f6da22d3-5ce5-4d5d-b0cd-240924e26928" condition="contains" preserve-case="false"
    negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
14            <Context document="RegistryItem" search="RegistryItem/Path" type="mir"/>
15            <Content type="string">\Software\Microsoft\Windows\CurrentVersion\Run</Content>
16          </IndicatorItem>
17          <IndicatorItem id="545932aa-3684-46dd-bc0d-5222e480563b" condition="matches" preserve-case="false"
    negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
18            <Context document="RegistryItem" search="RegistryItem/Text" type="mir"/>
19            <Content type="string">.*(AppData|Application Data)(\\Roaming)?\\[a-f0-9]{7}.exe</Content>
20          </IndicatorItem>
21          <IndicatorItem id="1f0390c2-5b92-4488-8abf-65d97016ed12" condition="contains" preserve-case="false"
    negate="false" group-id="cb6159bc-85de-4387-9a32-ff799c216687">
22            <Context document="FileItem" search="FileItem/PEInfo/ImportedModules/ImportedFunctions/string"
    type="mir"/>
23            <Content type="string">IsDebuggerPresent</Content>
24          </IndicatorItem>
25          <IndicatorItem id="ffa643ea-0b7a-418d-9e22-85669a048539" condition="contains" preserve-case="false"
    negate="false" group-id="7b4bfbc1-a135-45a0-82d9-801ce0e10e81">
26            <Context document="RegistryItem" search="RegistryItem/Path" type="mir"/>
27            <Content type="string">\Software\Microsoft\Windows\CurrentVersion\Run</Content>
28          </IndicatorItem>
29          <IndicatorItem id="a35d27e0-dd71-4f33-839a-a2b162ee728d" condition="matches" preserve-case="false"
    negate="false" group-id="7b4bfbc1-a135-45a0-82d9-801ce0e10e81">
30            <Context document="RegistryItem" search="RegistryItem/Text" type="mir"/>
31            <Content type="string">[A-Z]:\\[a-f0-9]{7}\\[a-f0-9]{7}.exe</Content>
32          </IndicatorItem>
33          <IndicatorItem id="8f451de5-9316-43f2-8804-d8f5a9861ed8" condition="contains" preserve-case="false"
    negate="false" group-id="7b4bfbc1-a135-45a0-82d9-801ce0e10e81">
34            <Context document="FileItem" search="FileItem/PEInfo/ImportedModules/ImportedFunctions/string"
    type="mir"/>
35            <Content type="string">IsDebuggerPresent</Content>
36          </IndicatorItem>
```

4.0 Summary:



We determined through device trajectory in AMP that the delivery method of this attack was a document labeled invoice (in Italian). The exploitation vector was through Microsoft Word's execution of macros in the document file.

We identified the Command & Control being used in the actively compromised host, and identified additional domains and URL paths through Threat Grid analysis.

Through analysis in Threat Grid we also identified the actions of the payloads.

We were then able to remediate further infections by creating a black list, and by identifying any other infections within the environment using Endpoint IOCs.

The created Endpoint IOC is available in its entirety here:

http://immunet-janus-helpdoc.s3.amazonaws.com/Sample%20IOCs/CryptoWallDropper.ioc