The Middle East under Malware Attack
Dissecting Cyber Weapons

Sami Zhioua
Information and Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
zhioua@kfupm.edu.sa

***Abstract*—The Middle East is currently the target of an unprecedented campaign of cyber attacks carried out by unknown parties. The energy industry is praticularly targeted. The attacks are carried out by deploying extremely sophisticated malware. The campaign opened by the Stuxnet malware in 2010 and then continued through Duqu, Flame, Gauss, and Shamoon malware. This paper is a technical survey of the attacking vectors utilized by the three most famous malware, namely, Stuxnet, Flame, and Shamoon. We describe their main modules, their sophisticated spreading capabilities, and we discuss what it sets them apart from typical malware. The main purpose of the paper is to point out the recent trends infused by this new breed of malware into cyber attacks.**

***Keywords*-Malwares; Information Security; Targeted Attacks; Stuxnet; Duqu; Flame; Gauss; Shamoon**

I. Introduction

Malicious software, or malware, plays a part in most computer intrusion and security incidents. Any software that does something that causes harm to a user, computer, or network can be considered malware including viruses, trojan horses, worms, rootkits, and spyware [1]. Malwares are emerging as the threat of the future. Indeed, in 2011 more than 3 out of 4 attacks resulting in financial losses were due to malware infection [2]. Malwares are getting more and more sophisticated. The recent campaign of malware-based attacks targeting the middle east is a manifestation of this trend. Several organizations in the middle east, in particular in the energy industry, reported recently infections with sophisticated malware which exhibit suspicious similarities. The first one to be discovered was Stuxnet [3], [4] which is the first malware targeting specifically critical infrastructure systems (e.g. Supervisory Control and Data Acquisition (SCADA) systems, nuclear power plants, etc.). Stuxnet attack on Natanz uranium enrichment facilities is believed to be the main reason of the (at least) 3 years delay of Iran's nuclear program [5]. Duqu trojan [6] was revealed in September 2011 and was designed mainly for extremely targeted espionnage activity. Duqu shares a lot of code with Stuxnet and there are several technical evidences that they have been designed by the same unknown entity [1]. The next malware of this lineage was Flame [7] which was discovered in May 2012 by Kaspersky Lab while investigating another piece of malware called Wiper [8]. Flame features very unusual characteristics such as large size, large number of modules, self adapting, etc. As Duqu, Flame's objective is data collection and espionnage. Gauss [9] is another data stealing malware discovered in June 2012 by Kaspersky Lab focusing on banking information. Flame and Gauss exhibit striking similarities and several technical evidences indicate that they come from the same "factories" that produced Stuxnet and Duqu [9]. The latest malware-based attack targeting the middle east was the Shamoon attack on Saudi Aramco [10]. Shamoon malware [11] is less sophisticated than the previously mentioned malware and there are technical evidences that it is the work of amateurs. Shamoon's main objective is to wipe data from Windows computers and then to tamper with the Master Boot Record (MBR) of the storage media, making the computer inaccessible. Shamoon resulted in the complete destruction of the content of around 30,000 workstations in Saudi Aramco [12], [10].

Given the amount of effort required to build these sophisticated malware and the spectacular consquences of the attacks, one can draw at least two conclusions. First, the unknown parties behind them are not typical cybercriminals or hacktivists. Second, these malware are using state-of-art hacking techniques. This paper gives a technical overview of the main hacking techniques utilized by these malware. The paper is a result of an extended survey of a large number of technical reports as it rehashes available reports focusing on the key features of the malware and the new hacking techniques. We consider carrying out this exercise essential to understand the recent hacking trends and hopefully to forecast future attacks.

The next five sections are dedicated respectively to Stuxnet, Flame, and Shamoon malware. Section V outlines the common and relevant trends in these malware. Finally Section VI concludes.

[1]Several speculations exist in the media about the origin of this cyber attacks campaign. In this paper, we focus only on the technical characteristics of these malware.

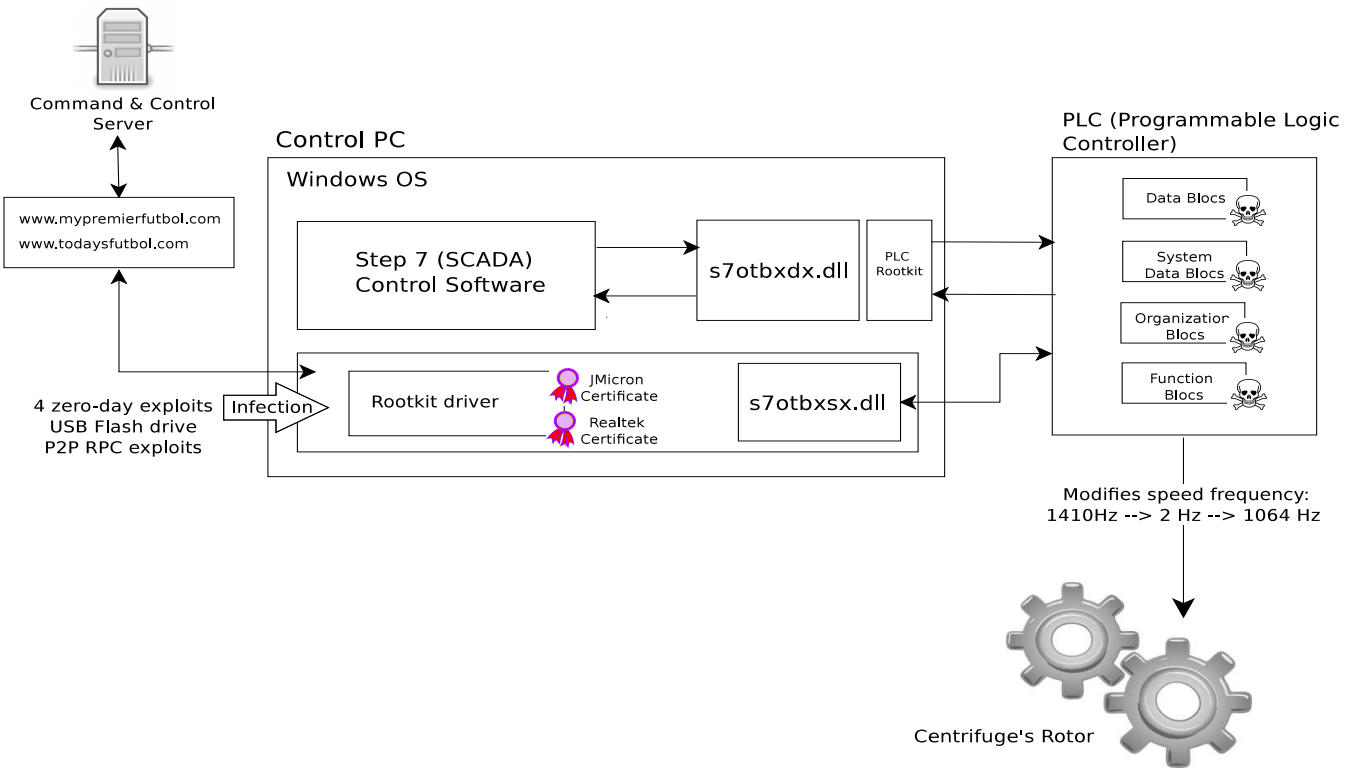

Figure 1. Overview of Stuxnet Malware Operation.

II. Dissecting Stuxnet Malware

Stuxnet is a malware targeting nuclear facilities, in particular the Natanz uranium enrichment facility in Iran [13]. The main goal of Stuxnet is to reprogram the Programmable Logic Controller (PLC) of the facility such that to physically harm the attached hardware equipment while hiding the malicious activity from the operator of the equipment. The Stuxnet malware attack operates at three levels:

1) compromising Windows operating system
2) compromising Step 7 (SCADA) software application
3) compromising the Programmable Logic Controller (PLC).

Figure 1 is an overview of how Stuxnet operates.

A. Compromising Windows

The main objective of Stuxnet is to tamper with PLC functionalities. PLC is a small computer system that operates in real time and plays the role of interface between the software application (Step 7) and the industrial physical machines. A PLC is not a typical computer with an operating system and an internet connection. The plot behind Stuxnet is to compromise PLC through compromising the computer used by the engineer in charge of reprogramming the PLC. To reprogram the PLC, the engineer should connect his computer (Windows) to the PLC with a data cable. Once the PLC is configured, the Windows computer can be unplugged and PLC will function by itself.

Self-propagation Stuxnet can distribute itself using an unprecedented set of four zero-day exploits, namely, MS10-046 [14], [15], MS10-061 [16], MS10-073 [17], and MS10-092 [18]. zero-day exploits are very valuable attack vectors that are very difficult to find and hence are not typically wasted in everyday malware. The use of four zero-day expoits in a single malware is another indicator of the unusual sophistication and cost of Stuxnet.

The main technique used by Stuxnet to distribute itself is the LNK exploit (MS10-046) delivered in a USB drive. The vulnerability exists because Windows incorrectly parses shortcuts (.LNK files) in such a way that malicious code may be executed when the icon of a specially crafted LNK file is displayed. A non-patched Windows OS will be compromised as soon as Windows Explorer is used to open the USB drive containing the malicious LNK files[2].

In order to replicate itself over the network, Stuxnet uses a print spooler service vulnerability (MS10-061). This attack targets Windows machines with file and print sharing turned on. It proceeds by sending a specially crafted print request

[2]A typical configuration of the USB drive will contain several LNK files each one for a particular Windows OS version (e.g. XP, Vista, 7, Server 2003).

of two documents. Due to a flaw in the print spooler, the documents can be printed to files in the Windows %*system*% directory. Then, under certain conditions, the first file (*sysnullevnt.mof*) will be used to register providers and events and also to launch the second file (dropper: *winsta.exe*) whose execution results in the infection of the system.

Windows Rootkit Stuxnet has rootkit capabilities to hide files and to inject code into processes. It drops two Kernel-mode drivers during installation. These drivers are not encrypted nor packed. However, they are signed by two legitimate certificates stolen from separate companies, JMicron and Realtek both with offices in Hsinchu Science Park, Taiwan [4]. The signing of drivers allowed to install the rootkit drivers succesfully and allowed Stuxnet to remain undetected for a long period of time.

Command and Control Server Communication Once a Windows system is compromised with Stuxnet, it tries to communicate with the Command and Control (C&C) Server. First, it checks whether an internet connection is available by trying to open *www.windowsupdate.com* and *www.msn.com*. Then it communicates with the C&C server using the domains *www.mypremierfutbol.com* and *www.todayfutbol.com*. In one direction, the C&C server sends updates and binary codes to be executed for the Stuxnet client while in the other direction, Stuxnet client will upload system information about the victims and the installed Industrial Conrol Systems softwares to the server.

B. Compromising Step 7 Software Application

Once a machine is infected, Stuxnet will hook specific APIs used to open Step 7 projects. As soon as such project is opened, the path to the project's folder is recorded and the folder is infected. Loading any Step 7 project in an infected folder causes Stuxnet to execute. Contaminating a Step 7 project consists in dropping several DLL files in different system folders. However, the main action carried out by Stuxnet is to rename the original *s7otbxdx.dll* file by *s7otbxsx.dll* and then provides its own compromised version of *s7otbxdx.dll*.

The *s7otbxdx.dll* is a library file used by Step 7 software to communicate with the PLC. The dll file exports several routines to read and write code blocks to/from the PLC. By replacing the orginal version of *s7otbxdx.dll* by its own compromised version, Stuxnet can intercept any communication between Step 7 software and the PLC.

C. Compromising PLC

Stuxnet is a highly targeted malware. Although its main goal is to attack PLC, not any PLC will trigger Stuxnet damaging payload. Stuxnet will be triggered only if the PLC uses a Profibus communications processor[3]. Moreover, Stuxnet will only launch the damaging payload if the PLC is using one of two frequency converter drives: one manufactured by an Iranian company and one by a Finnish company. The only known site using this exact configuration for PLC is the Natanz uranium enrichment facility which is the only site with reported Stuxnet related damage.

The damaging payload of Stuxnet consists in the following. It keeps monitoring the Profibus messaging bus. When a certain condition is met (e.g. the attached system is spinning with frequency between 807Hz and 1210Hz) it modifies the frequency to 1410Hz then to 2Hz then to 1064Hz. The intended consequence of this manipulation is that the stresses from the excessive, then slower, speeds cause the aluminium centrifugal tubes to expand forcing parts of the centrifuges into excessive contact leading to the destruction of the machine [19].

In order to keep the infected code blocks on the PLC undetected, Stuxnet uses a PLC rootkit. The rootkit is contained in the fake *s7otbxdx.dll* library. Anytime a request from the Step 7 software application tries to access an infected block in the PLC, the request is intercepted and modified so that Stuxnet infected blocks are not discovered nor modified. To hide its malicious activity, Stuxnet records previous and normal operating frequencies and then feeds them to the PLC operator as well as the digital safety system[4]. Hence, while Stuxnet malicious payload is executing, everything appears normal to the PLC operators and the digital safety systems.

III. Dissecting Flame Malware

Flame malware has been discovered in late May 2012 however there are several evidences it is active in the wild as early as February 2010 [7]. Flame is a sophisticated attack platform with uncommon features such as the large size (900KB for the bare-bone version and 20MB when fully deployed), the use of a LUA virtual machine, the use of bluetooth and above all the ability to steal data in so many different ways.

The following sub-section is dedicated to a description of the basic modules of Flame. These modules carry out functionalities such as data collection, spreading, bluetooth espionage, escaping security solutions, etc. Then, in the next sub-section we show details about the infrastructure used by the attackers to command and control the infected systems.

A. Flame Modules

Flame's main file is called *mssecmgr.ocs*. Many parts of Flame modules are written in Lua. They are then interpreted through the Lua virtual machine. The presence of the Lua

[3]Profibus is a standard industrial network bus used for distributed I/O. It is a standard to link PLC to the physical devices.

[4]Digital safety systems are needed when a human operator cannot act quick enough in critical situations.

interpreter and the fact that the modules are written in Lua makes it very easy to extend the functionalities of the malware by other modules downloaded from the attack center. This is one of the important design features of Flame and what distinguishes it from typical malware. This capability to download and update modules improves significantly the efficiency of modules such as data collection and security products escaping.

EUPHORIA As mentioned above, Flame has worm capabilities and hence can spread from one system to another. The spreading techniques are implemented in several modules and include:

- The use of network shares
- With USB using a malicious autorun.inf file (used also by Stuxnet)
- Using zero-day exploits, in particular the LNK (MS10-064) exploit [14] which is the main spreading infection mechanism used in Stuxnet.

EUPHORIA is the module that controls the spreading mechansim via USB sticks. All these spreading techniques are not new since they were used by other malware, in particular Stuxnet. Flame however implements a brand-new spreading technique illustrated in the next paragraphs.

FLASK, JIMMY, and MICROBE The main objective of Flame Malware is to steal data from infected systems [20]. The three modules FLASK, JIMMY, and MICROBE are in charge of gathering enormous amount of information about the victims [21]. The first one, FLASK, is an info-stealer that collects a large amount of data available on an infected system. It records information such as the computer name, the OS version, the list of volumes, open TCP/UDP connections, cookies in web browsers, etc. JIMMY is a file scanner and leaker. It scans the storage media of the infected system for relevant files such as docx, ppt, csv, dwg. As of MICROBE, it is in charge of recording audio from existing hardware sources (microphone, etc.). It stores device configuration about multimedia devices and selects suitable recording devices. These modules use intelligence to collect relevant data: not all data is uploaded to the C&C server. Instead, Flame initially collects some preliminary information (metadata, summary, header info, etc.) which is sent to the attack center for analysis. In the light of that information the attacker decides about which files are juicier and instruct the Flame client to steal data accordingly.

BEETLEJUICE Flame is the first Windows malware using bluetooth [22]. The bluetooth functionality is encoded in the BEETLEJUICE module. When launched, this module enumerates devices around the infected machine and turns itself into a "beacon", that is, announces itself as a discoverable device. This bluetooth functionality allows the attacker to

- identify the victim's social networks,
- identify the victim's physical location,
- enhance information gathering (steal address book, SMS messages, exfiltrate already-stolen data through bluetooth connected devices which will bypass firewall and network controls[5].).

adventcfg.ocx Typical malware written by cyber-criminals and hacktivists use known techniques to bypass security products (firewalls, antivirus tools, intrusion detection systems). These techniques include encryption, obfuscation, anti-debugging, anti-reverse. Flame does not use any of these techniques. Instead, it chooses to "not disturb" the security solutions. For instance, if Flame client suspects that an attack will be detected if launched, the attack will not be launched at all. Espionage is moving slowly with lot of delay, which makes it hard to be noticed. *adventcfg.ocx* [23] is a module collecting data to improve the escaping capabilities of Flame. Whenever Flame notices that Windows OS is issuing a message or launching a tool that is referencing one Flame file or component, *adventcfg.ocx* takes a screenshot which is then sent to the C&C server for analysis. Based on this analysis the attacker will add code to improve the existing version for the next "update". This is one of the features that made Flame undetected for a long period of time (at least two years).

SUICIDE Shortly after its discovery by Kaspersky Lab, Flame went dark overnight [22]. Indeed, in the last week of May 2012, the C&C servers of Flame sent an updated command to all infected systems to completely delete itself. SUICIDE is the module in charge of committing suicide. It locates every file on disk, removes it, and overwrites the disk with random characters to prevent anyone from obtaining information about the infection. Since the triggering of the suicide operation, there were no reported active infections of Flame or other variants.

SNACK, MUNCH, and GADGET The spreading techniques mentioned earlier with the EUPHORIA module are known and have been already used by previous malware (Stuxnet, etc.). However, the main spreading mechansim of Flame is completely new and extremely sophisticated [24], [25]. The main idea is to take advantage of the automatic Microsoft's Windows update to infect other systems in the same network. This mechanism is implemented in the SNACK, MUNCH and GADGET modules. The first step is to carry out a Man-In-The-Middle (MITM) attack to redirect all victim's traffic through the Flame infected machine. When Internet Explorer (IE) is launched in a Windows

[5]It might happen that the infected system is in an protected/isolated network with no internet connection. This prevents direct internet connection with the C&C server. The infected system can use the nearby bluetooth devices as a bridge to communicate with the C&C server.

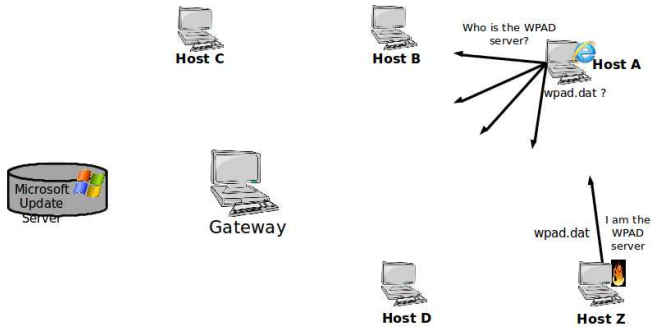

Figure 2. Flame Man-In-The-Middle Attack.

machine, IE starts by looking for a proxy configuration settings: it broadcasts a packet through the Web Proxy Auto-Discovery Protocol (WPAD) asking for the proxy settings (*wpad.dat*). SNACK module is in charge of sniffing the network for WPAD requests. Flame will then claim to be the WPAD server and provide a fake *wpad.dat* file[6]. Once the victim receives the fake *wpad.dat* file, it will set the Flame infected machine as a proxy for all its traffic.

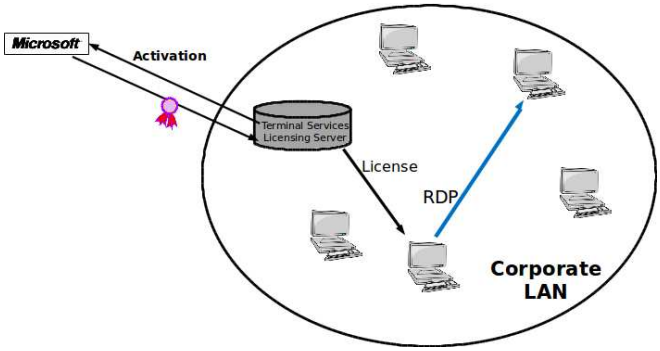

Figure 3. Leveraging Microsoft Certificate to Sign Code.

The next step is to intercept requests for Windows update. This is carried out by the MUNCH module. Once a Windows update request is intercepted, the GADGET module will prepare a fake update binary with the Flame installation file. It is widely known that Windows OS computers launch Windows update binaries without any restrictions provided that the update is genuine, that is, signed by a Microsoft certificate. Hence for this attack to work, the Flame client should sign the fake update with a Microsoft certificate. The trick to be able to sign code with a Microsoft certificate is illustrated in Figure 3. In a typical LAN, a client using Microsoft Terminal Services (or Remote Desktop Connction (RDP) to remotely access a Windows desktop needs a license. The licenses are typically issued by a Terminal Services Licensing Server (TSLS) which allows an entrprise to administrate and enforce licenses for connecting clients within its environment. Before using the TSLS, it must be activated by contacting Microsoft. Microsoft issues a limited use certificate allowing only to verify the ownership of the TSLS. Flame designers managed to use the certificate to sign code using a flawed signing algorithm [26], [27]. In June 3, 2012, Microsoft issued an security advisory [28] and the corresponding update to fix this issue by moving three certificates to the Untrusted Certificate Store making any code signed by them invalid. This is the first time that such certificate leveraging is used to spread malware.

[6]Normally, computer name resolution inside the network are carried out through a DNS server. However, if the DNS server does not have records registered, IE will use NetBIOS for name resolution. With NetBIOS every computer boradcasts its own name to identify itself. NetBIOS WPAD hijacking is a known MITM attack along with ARP Poisoning.

B. Flame Command and Control Servers

Two important features of Flame require infected machines to contact regularly a C&C server. First, the main goal of Flame is to steal data from victims. Infected machines need to upload regularly stolen data to the servers. Second, Flame is highly modular and constantly updated with new functionalities to gather data more efficiently and to improve its escaping capabilities. Infected machines need to contact regularly the servers for new updates and commands. Figure 4 shows how infected machine are connected to C&C servers and how these servers are controlled by the attack center.

When a computer is infected with Flame, it uses a default configuration of 5 domains to contact the C&C servers. Once it successfully connects to a server, the list is updated to reach around 10 domains. In total, the infected machines use 80 domains to contact the C&C servers [24]. These domains are registered with fake identities (with fake addresses mostly in Germany and Austria) and with a variety of registrars. All used domains point to a total of 22 C&C server IPs hosted around the world. These servers are hosted normally as any usual web server. The hosting companies providing the services are not aware of the activity of the servers. The attackers carefully configured the servers to look as any typical website server. These C&C servers are controlled by a single attack center.

Figure 5 shows the main components of the C&C and how it is used by the attack center to control the infected machines.

Server Setup The typical configuration of a Flame's C&C server is a Debian Linux virtual machine running under OpenVZ. It has a database (MySQL) and an Apache web server. This is the so-called LAMP (Linux, Apache, MySQL, PHP) setup. Due to the large number of deployed C&C servers, the admin at the attack center used automation to prepare the server environment. The admin connects to the server through ssh (port 22) and run some scripts, in particular `LogWiper.sh` which stops linux system logging daemons and deletes log files using `chkconfig`

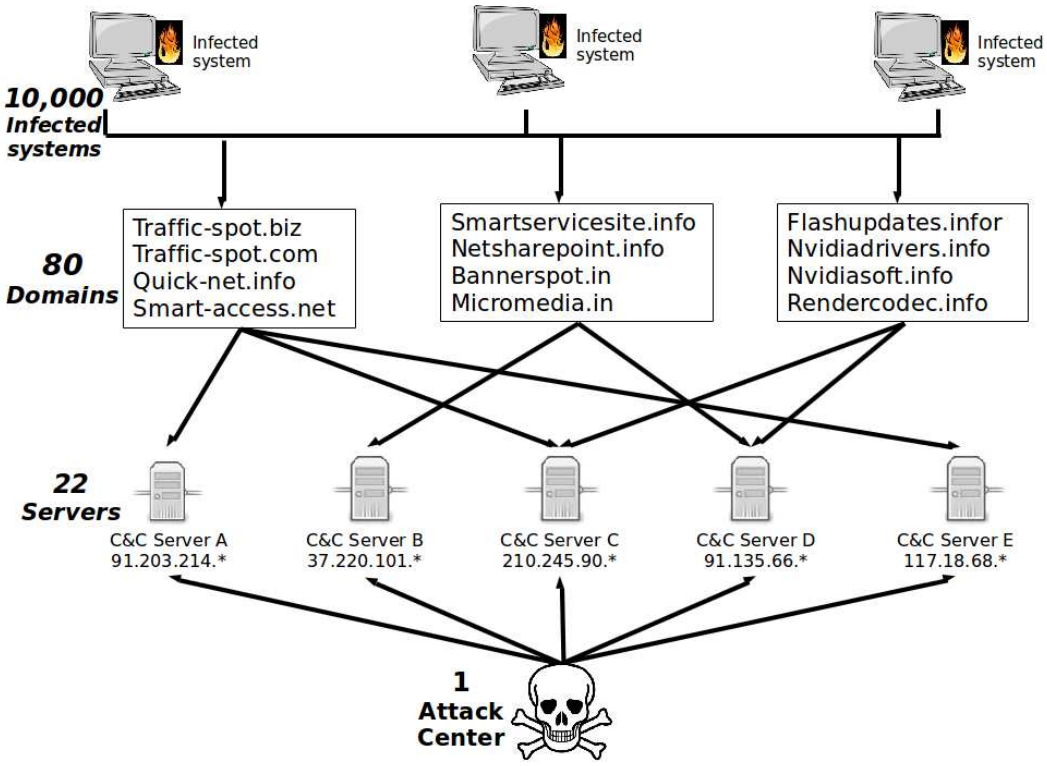

Figure 4. The Command and Control Platform behind Flame.

and `shred` commands which make any recovery operation impossible [29]. The script finally deletes itself. The admin schedules tasks that will periodically delete files and remove old entries from the database. For instance, stolen files from the infected machines are cleaned up every 30 minutes[7].

Data Flow The C&C server can be seen as a proxy between the attack center and the infected systems. In one direction, commands and updates are sent from the attack center to the infected systems. In the other direction, stolen data are sent from the infected systems to the attack center. This is achieved through the Apache web server at the C&C. Data flows in a military-like approach: one party uploads files on the server and then the other party will retrieve those files from the server. There is no direct communication between the attack center and the infected systems. The server has one important folder called `newsforyou` which in turn contains three sub-folders `ads`, `news` and `entries` whose roles are as follows:

- The `ads` folder is where commands and updates meant for a specific infected system are uploaded by the attack center.
- The `news` folder is where commands and updates for all infected systems are uploaded by the attack center.
- The `entries` folder is where stolen data is uploaded by infected systems and then to be retrieved by the operator of the attack center.

The operator of the attack center uses a GUI control panel to upload and download data from the C&C Apache server. Infected systems use *GET_NEWS* command to retrieve commands and updates and *ADD_ENTRY* command to upload stolen data. The amount of stolen data in one sample C&C server is 5.5GB for a period of one week [21]. The data stolen data is encrypted using a public key available on the server. The corresponding private key is only known by the attack coordinator in the attack center. Even the admin and operator do not know the private key and hence do not have access to the stolen data. This hierarchical structure at the attack center is another evidence that the attackers are not typical cyber-criminals or hacktivisits.

Database Given the large number of infected systems and the need to keep track of the commands and updates sent to each one of them, the server maintains a MySQL database. The database stores data about:

- Connecting clients
- Packages to send to the clients
- Encryption settings
- Authentication to access the control panel.
- Etc.

[7]Deleting the stolen files is done after the files are uploaded to the attack center.

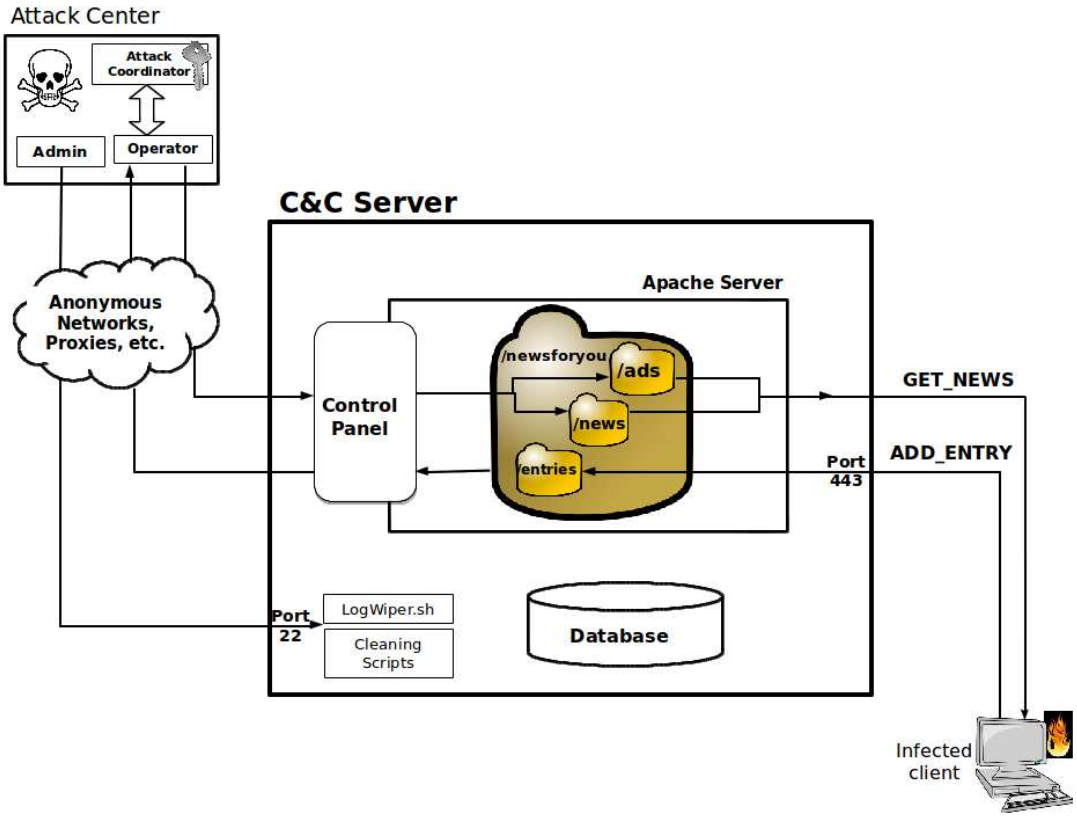

Figure 5. The Command and Control Server.

Flame malware is also equipped with features to steal data from protected environments. It is a common practice in organization to disconnect a sub-network from internet and to use it to manipulate confidential data. This makes these protected environments an important target of spying malware such as Flame. In order, to steal data from these protected environments, Flame uses a hidden database loaded in USB sticks. If a USB stick is inserted into an infected system in such environments, Flame reads the hidden database (if it does not exist, it will create one), and checks if the USB stick has already been in a computer with an internet connection. If it is the case, Flame begins storing leaked documents in the hidden database.

As mentioned in the previous section, the SUICIDE module of Flame has been activated as soon as the first reports about the discovery of the malware are published (May 2012). However, according to analyzed samples, Flame clients (*CLIENT_TYPE_FL*) constitute only one out of four types of infected clients (*CLIENT_TYPE_SP*, *CLIENT_TYPE_SPE*, and *CLIENT_TYPE_IP* being the others) [29]. This indicates that the attackers behind Flame can deploy new variants anytime.

IV. Dissecting Shamoon Malware

Shamoon (known also as Disttrack) is a recent malware used in a targeted attack against at least two organizations in the energy sector in the middle east [30]. It is not a typical malware in that its goal is mainly to carry out the maximum destruction possible. Indeed, instead of staying under the radar and collect information (financial, passwords, etc.), Shamoon was designed to overwrite and wipe the files and the Master Boot Record (MBR) of the computer making it unusable. Compared to Flame, Shamoon is less sophisticated and from the analysis of the malware samples, there are evidences that the attackers behind it are simply amateurs.

Figure 6 shows the components of Shamoon's main file called *TrkSvr.exe*. The file is a 900KB Portable Executable (PE) file with a number of encrypted resources. The encryption routine is a simple Xor cipher. The main components of the file are the dropper (installs the malware and drops the other modules), the wiper (in charge of the erasing) and the reporter (reports infection information back to the attacker). The last encrypted resource is 64 bits version of the malware.

A. Dropper

The dropper component is not encrypted and is directly retrieved from the main malware file. It is in charge of installing the malware, dropping the other components and

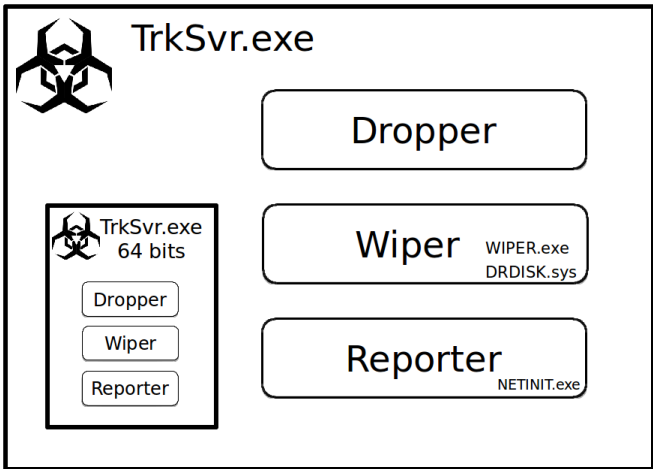

Figure 6. Shamoon Malware Components.

launching the service. When executed it performs the following actions:

- Copies itself to the Windows *%system%* folder.
- Drops the reporter component (after decrypting the corresponding resource) in the *%system%* folder under *netinit.exe*.
- drops the wiper component in the same *%system%* folder. The name for the dropped wiper executable file is chosen randomly among a set of fixed names (e.g. *caclsrv, fsutl, ntx*, etc.).
- Tries to infect other systems in the same LAN by attempting to copy itself in windows shared folders of targets.
- Creates a task to execute itself.
- Creates a TrkSvr service to start itself whenever windows starts.

Hence, the main spreading technique of Shamoon is through network shares [30]. Once a target is found, the malware will attempt to remotely open and close a list of files to determine if it has acces. If it has access it will copy and execute itself using *psexec.exe*.

B. Wiper

The most interesting component of Shamoon malware is the wiper which in charge of the destructive tasks. It is dropped and executed only after a hardcoded date is reached (August 15, 2012, 08:08 UTC in the case of Saudi Aramco attack). The wiper will wipe the following in order [31]:

1) A list of files, in particular files within folders containing the following names: download, document, picture, music, video, desktop.
2) Master Boot Record (MBR).
3) Active partition.

The list of file names to be overwritten are first collected and written in *f1.inf* and *f2.inf* files. The files are then overwritten with a JPEG image representing a burning USA flag. But due to a coding mistake, the files are overwritten only by the small upper part of the JPEG image [32]. This is one indication that the attackers behind it are simple amateurs. The next action of the wiper is to overwrite the MBR. However, tampering with the MBR is not allowed for user-mode applications. The workaround used by Shamoon to bypass this problem is to overwrite the existing disk driver with another legitimate driver (*DRDISK.sys*) signed by Eldos company [33]. The new driver enables user-mode applications to read and write to disk sectors. This is yet another example showing that cybercriminals and malware developers are always searching for covert ways to access a system's kernel. Legitimate and signed drivers that can be exploited for malicious activities are very demanded in the hacking community.

C. Reporter

The reporter component is responsible for sending infection information back to the attacker. In particular, the infected system will send the content of *f1.inf* file which is already filled with the names of files to be overwritten. Information is sent in a typical HTTP GET request. It includes the follwing information:

- The domain name of the infected system
- A number specifying how many files were overwritten
- The IP address of the infected system
- The *f1.inf* file

Shamoon malware is less sophisticated than Flame, Stuxnet, Duqu, etc. However, the attackers behind it could carry out a large-scale attack on a large organization.

V. Recent Malware Trends

Several common trends are emerging from these successful recent malware attacks showing once again the infeciency of the current security mechanisms. It is important to point out and discuss these trends in order to understand where the security field is heading.

A. Sophisticated Malwares

The most striking characteristic of the described malware is their sophistication. This sophistication is a result of a significant amount of effort and resources spent in the development of the malware. There is a wide agreement that only teams of developers working for long period of times (several months if not years) could produce such advanced attack toolkits:

- Most of the malware are using one or several zero-day exploits while finding such exploits requires a significant effort and time from security and hacking experts. Zero-day exploits are highly demanded in the hacking communities and can be sold with prices reaching 6 figures [34].

- Developing a malware such as Stuxnet to reach and tamper with a PLC in a highly protected zone requires to reconstruct the same configuration and setting (software and probably some hardware equipment) at the target site for testing purposes.
- Leveraging limited certificates to sign binaries, as in the Flame attack, can only be done by very knowledgeable cryptographers.
- The large number of registered domains and dedicated/compromised servers in Flame, Gauss, and Duqu attacks requires significant resources to setup and maintain.

B. Targeted Malwares

Recent malware are directed towards specific targets. The spreading mechanism is often controlled. The goal is not to infect the maximum number of victims. It is rather to infect a specific set of targets. At the same time, this contributes in keeping the malware undetected and working under the radar for a longer period of time. A targeted malware is a bigger threat to networks than mass malware, because it is not widespread and security products will not be able to provide a timely protection against it.

C. Certified Malwares

A common feature of all described malware is the use of legitimate certificates to sign parts of their malicious code. In the case of Stuxnet and Duqu, stolen certificates have been used to sign drivers allowing a stealth installation. In the case of Flame, a limited certificate has been leveraged by the attackers to sign malicious binary code exploiting a flaw in an old signing algorithm. In the case of Shamoon, a third-party certified disk driver has been used as is to allow a user-level application to write on the Master Boot Record. Bencsath and Pek [6] provide a relevant discussion about using certificates to sign malicious code.

D. Modular Malwares

Malwares commanded by a C&C server are common in cyber attacks. However, malware extending their capabilities while deployed in the victim is quite new and has been extensively used in the recent cyber attacks campaign on the middle east[8]. Downloading additional modules and executables after deployment allows to fine-tune the attack as more appropriate features are used for specific targets. Duqu malware used an extreme version of this feature as new modules are compiled and built specifically for every new infection. This feature allowed Flame to remain undetected for a long period of time as the module in charge of escaping security products was continuously updated by downloading regularly improved versions of the module.

[8]Even Shamoon malware had that capability implemented but it was not used in the Saudi Aramco attack as the goal was primarly to completely erase infected systems.

E. USB Spreading Malwares

USB drives, in addition to zero-day exploits, are emerging as the main infection vector in targeted attacks. Crafting a USB drive with a malicious LNK or autorun.inf file has been extensively used in particular in the initial infection in a LAN. In Flame attacks, USB drives were used to steal information from victims in protected zones (without internet connection). In the case of Stuxnet, there are two different speculations for the initial infection vector of the Natanz uranium enrichment facility and both of them involve USB drives. The first theory suggests that the malware has been spread via flash drives distributed at a SCADA conference [4]. The second theory suggests that the malware got inside the facility through a Russian integrator company that built the plant where one of the engineers was lured to accept an infected USB drive which infected his laptop and then the PLC which he is manipulating [13].

F. Suiciding Malwares

All described malware (except Shamoon) have an uninstallation module. The module completely removes the malware from a system, deleting every single trace of its existence. The module can be launched remotely by the attack center (through the C&C servers) at any time. Due to the amount of effort required in the developement of these advanced malware, the attackers commanded the malware to commit suicide (both at infected machines and at the C&C servers) as soon as it is detected and related details posted online. This makes any forensics investigation very difficult to carry out and leaves the possibility to deploy another variant of the malware in future attacks.

VI. Conclusion

The middle east, in particular the energy industry is the target of a cyber attacks campaign. Some unknown attackers are deploying very sophisticated and targeted malware. In this paper we illustrated the details of the most famous malware used in this campaign, namely, Stuxnet, Flame, and Shamoon. Stuxnet is a malware targeting exclusively the PLC of Natanz uranium enrichment facility in Iran and deployed with an unprecedented set of four zero-day exploits. Flame is an extremely sophisticated malware whose goal is to intelligently steal interesting data with various techniques (office files, audio recording, bluetooth, etc.). Its large size, modularity, security solutions espacing tricks, and new spreading techniques make it one of the most sophisticated e-threats ever deployed. Shamoon is a relatively simple but very destructive malware whose main feature is the use of a legitimate and signed disk driver to be able to overwrite the Master Boot Record (MBR) of the infected systems making them unusable.

This new breed of malware is infusing new trends in cyber attacks. Most of these malware are unusually sophisticated,

directed towards specific targets, using stolen digital certificates, modular with self-updating capabilities, increasingly using USB drives and zero-day exploits to spread, and ready to commit suicide at any moment.

The spectacular "Success" of these malware attacks as well as their sudden vanishment are clear warnings of other waves of similar attacks in the future.

REFERENCES

[1] M. Siroski and A. Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. San Francisco, CA, USA: No Starch Press, 2012.

[2] C. Henderson, "2011 global security statistics and trends," 2011.

[3] N. Falliere, L. Murchu, and E. Chien, "W32.stuxnet dossier," Symantec Security Response, Tech. Rep., February 2011.

[4] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet Under the Microscope (Revision 1.31)," ESET, Tech. Rep., 2011.

[5] W. Broad, J. Markoff, and D. Sanger, "Israeli Test on Worm Called Crucial in Iran Nuclear Delay," "www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html", 2011.

[6] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "Duqu: Analysis, detection, and lessons learned," in *ACM European Workshop on System Security (EuroSec)*. ACM, 2012.

[7] A. Gostev, "The flame: Questions and answers," Kaspersky, Tech. Rep., May 2012.

[8] GReAT: Kaspersky Lab Expert, "What was the wiper thing?" Kaspersky, Tech. Rep., 2012.

[9] ——, "Gauss: Abnormal Distribution," Kaspersky, Tech. Rep., August 2012.

[10] BBC News, "Shamoon virus targets enery sector infrastructure," "http://www.bbc.co.uk/news/technology-1929379", 2012.

[11] Symantec Security Response, "The shamoon attacks," Symantec, Tech. Rep., August 2012.

[12] John Leyden, "Hack on Saudi Aramco hit 30,000 workstations, oil firm admits," "http://www.theregister.co.uk/2012/08/29/", 2012.

[13] Steven Cherry, "How Stuxnet is Rewriting the Cyberterrorism," "http://spectrum.ieee.org/podcast/telecom/security/how-stuxnet-is-rewriting-the-cyberterrorism-playboo", 2010.

[14] Microsoft, "Microsot Security Bulletin MS10-046: Vulnerability in Windows Shell Could Allow Remote Code Execution," 2010.

[15] ——, "Microsot Security Advisory CVE-2010-2568: Vulnerability in Windows Shell Could Allow Remote Code Execution," 2010.

[16] ——, "Microsot Security Bulletin MS10-061: Vulnerability in Windows Print Spooler Service," 2010.

[17] ——, "Microsot Security Bulletin MS10-073: Vulnerability in Windows Kernel-Mode Drivers," 2010.

[18] ——, "Microsot Security Bulletin MS10-092: Vulnerability in Windows Task Scheduler," 2010.

[19] H. Stark, "Stuxnet Virus Opens New Era of Cyber War," "http://www.spiegel.de/international/world/mossad-s-miracle-weapon-stuxnet-virus-opens-new-era-of-cyber-war-a-778912.htm", 2011.

[20] Symantec Security Response, "W32.flamer: Enormous data collection," Symantec, Tech. Rep., June 2012.

[21] bbotezatu (BitDefender), "Cyber Espionage Reaches New Levels with Flamer," BitDefender, Tech. Rep., 2012.

[22] Symantec Security Response, "Flamer: Urgent suicide," Symantec, Tech. Rep., June 2012.

[23] bbotezatu (BitDefender), "Flamer Used QA Module to Thwart Antivirus," BitDefender, Tech. Rep., 2012.

[24] A. Gostev, "Flame: Replication via Windows Update MITM Proxy Server," Kaspersky, Tech. Rep., June 2012.

[25] Symantec Security Response, "W32.flamer: Microsoft windows update man-in-the-middle," Symantec, Tech. Rep., June 2012.

[26] ——, "W32.flamer: Leveraging microsoft digital certificates," Symantec, Tech. Rep., June 2012.

[27] D. Goodin, "Flame malware was signed by rogue Microsoft certificate," "http://arstechnica.com/security/2012/06/flame-malware-was-signed-by-rogue-microsoft-certificate", 2012.

[28] Microsoft, "Microsoft Security Advisory (2718704): Unauthorized Digital Certificates Could Allow Spoofing," 2012.

[29] GReAT: Kaspersky Lab Expert, "Full analysis of flame's command & control servers," Kaspersky, Tech. Rep., September 2012.

[30] D. Tarakanov, "Shamoon the wiper in details," Kaspersky Labs, Tech. Rep., August 2012.

[31] ——, "Shamoon the Wiper: Further Details (Part II)," Kaspersky Labs, Tech. Rep., September 2012.

[32] Symantec Security Response, "The shamoon attacks continue," Symantec, Tech. Rep., September 2012.

[33] "Eldos corporation," "http://www.eldos.com".

[34] A. Greenberg, "Shopping For Zero-Days: A Price List for Hacker's Secret Software Exploits," "http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits", 2012.