THE UNBEARABLE LIGHTNESS OF APTING

Yaniv Balmas, Ron Davidson & Shahar Tal
Check Point Software, Israel

Email {yanivb, rond, shahartal}@checkpoint.com

ABSTRACT

APT campaigns are typically described with awe surrounding the technical achievements enabled by the level of resources and capacity conceivably available only to nation-state governments and intelligence agencies, often dubbed APT groups. These reports contribute to the perception of a very high technological barrier-to-entry to the advanced targeted campaign 'market'.

This paper will present in detail our investigation of a carefully orchestrated targeted campaign that had been active since 2012 until interrupted by our Threat Research operation in 2015. This attacker group was observed employing several attack techniques, exploiting vulnerabilities and notably operating a custom-made malware implant code named Explosive.

As the investigation unfolded, our researchers collected evidence of this campaign successfully infiltrating many organizations, with a target distribution that strongly suggested a nation-state/political group interest.

We were surprised to find that the Volatile Cedar campaign has been active for almost three years, simply because many of its technical aspects cannot be considered 'cutting edge'. The campaign, however, has been operational continually and successfully throughout this entire timeline, evading detection by the majority of AV products. This success is due to a well planned and carefully managed operation that constantly monitors its victims' actions and responds rapidly to detection incidents.

INTRODUCTION

Volatile Cedar is a highly targeted and very well-managed campaign. Its targets are chosen carefully, confining the infection spread to the bare minimum required to achieve the attackers' goal while minimizing the risk of exposure. Our analysis leads us to believe that the attackers conduct a fair amount of intelligence gathering to tailor each infection to its specific target.

The campaign's initial targets are mostly public web servers running the *Windows* operating system. We believe this is because these machines serve as publicly exposed, easily accessible gateways to private and more secure internal networks. As these servers have common business functionality, their security is often sacrificed for productivity, making them an easy target for attackers. Once an attacker gains control over these servers, he can use them as a pivot point to explore, identify and attack additional targets located deeper inside the internal network.

A typical Volatile Cedar attack begins with a vulnerability scan of the target server. Once an exploitable vulnerability is located, it is used to inject a web shell code into the server. The exploited vulnerabilities are mostly commonly known vulnerabilities – no 0-days have been identified in our investigation.

The web shell is then used by the attacker to control the victim server and is the means through which the Explosive trojan is implanted into the victim server. This trojan allows the attackers to send commands to all targets via an array of C&C servers. The command list contains all the functionality required by the attacker to maintain control and extract information from the servers and includes keylogging, clipboard logging, screenshots, run commands, etc.

Occasionally – mostly in cases where large data extractions are required – the attacker sets up additional SSH tunnels connecting to the attacker-controlled servers.

ATTACK TIMELINE

The first evidence of Explosive was detected in November 2012. Over the course of the timeline, several versions have been detected. New version release dates appear to be closely related to the occurrence of an AV detection event on the previous version, a fact which emphasizes the efforts taken to conceal the attack.

The latest Explosive version was released in June 2014 and is still active at the time of writing this paper. See Figure 1 for more details of the timeline.

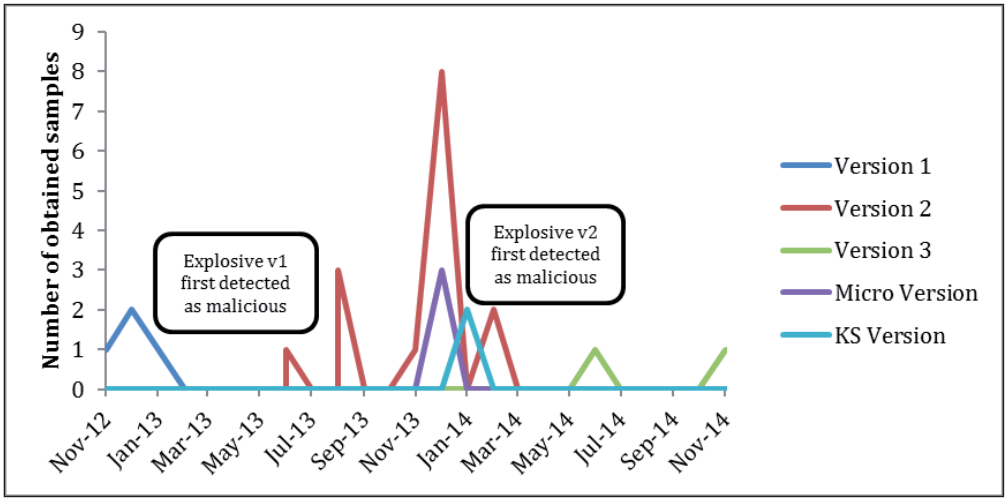

Figure 1: Explosive timeline.

STEALTH

The Explosive trojan goes to a lot of effort to hide from common detection tools and merge into its surroundings.

It avoids AV detections by frequently checking AV results and by changing versions and builds on all infected servers when any traces of detection appear.

New versions of the malware are equipped with a dedicated thread to monitor memory consumption to prevent common server administration utilities from detecting the Explosive processes. Once Explosive's memory consumption reaches a predefined threshold, its hosting process is immediately restarted.

API activities which may be considered suspicious are detached from the main logic file and contained in a separate DLL. This enables the attackers to make sure that heuristic detections do not lead to exposure of the trojan logic itself.

Custom configurations are set on a per target basis. For example, each trojan configuration contains periods of 'radio silence' during which Explosive does not initiate any network communication. These periods are set according to the specific target's working hours and low traffic periods.

A dedicated thread makes periodic 'secure checks' with the C&C server to confirm that it is safe to operate. Once the response to these checks is negative, the Explosive trojan ceases all operations until instructed otherwise.

These stealth techniques lead us to believe that whomever is responsible for this malware is probably very well organized, calculated and determined to sustain the attack for a long period of time.

CONTROL NETWORK

The campaign uses a multi-tiered server backend framework to control the targeted systems. This backend framework is composed of three major tiers:

- **Tier 1 – C&C servers**: The Explosive trojan attempts to connect to its C&C servers, which are used to send commands and receive information extracted from the targets. Each Explosive version has a default hard-coded C&C address. Different versions use different C&C servers.
- **Tier 2 – Static update servers**: These servers are periodically connected to obtain the current C&C address. If a new C&C address is available, the default C&C server is updated with the new one. The static C&C updater address is also hard coded as part of the Explosive configuration section.
- **Tier 3 – Dynamic update servers**: If the static C&C server is non responsive, the Explosive infection initiates a custom DGA algorithm which attempts to connect to the dynamic update servers. Once connected, these servers operate in the same way as the static updaters. Some Explosive versions also use the dynamic update servers as their C&C servers.

The server framework is diverse. While some servers are owned (and possibly also hosted) by the attackers, others use publicly shared hosting frameworks or even compromised legitimate servers.

INFECTION SPREAD

Evidence shows that the Explosive trojan leverages its

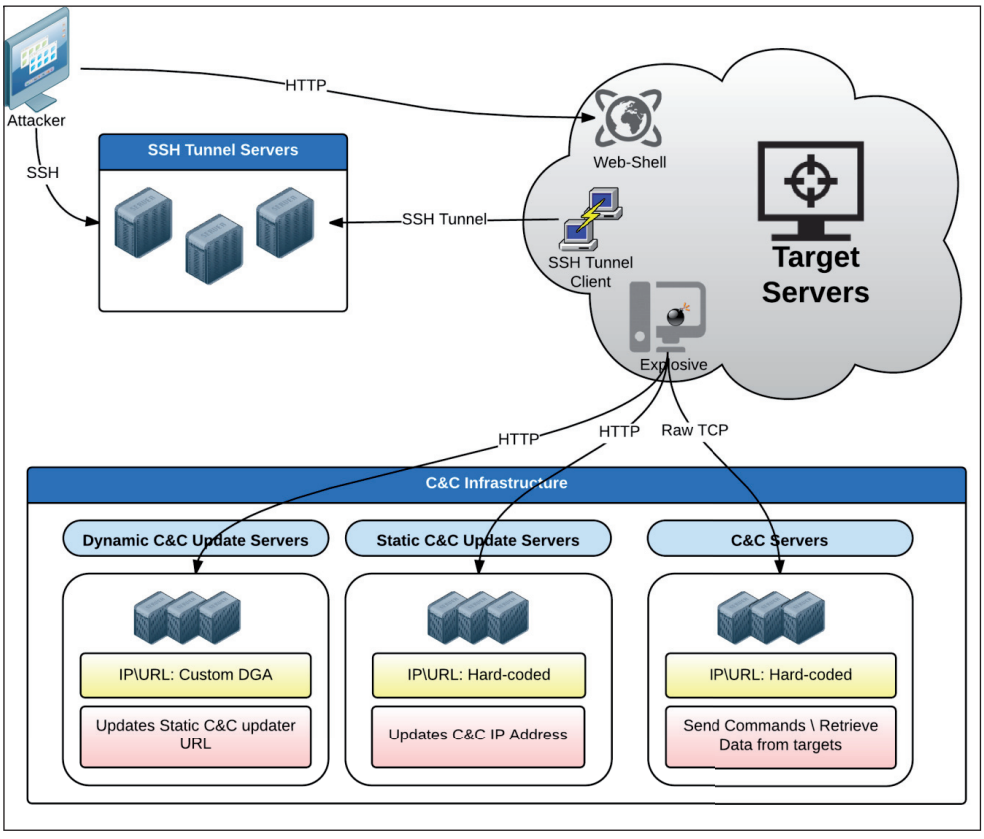

Figure 2: Control network.

keylogging capabilities to gain access to administrator passwords entered on the target servers. Additionally, residues of custom-built port scanners and several other attack tools have been found on the victim servers, leading us to believe the attackers use the initially infected servers as a pivot to spread manually to the entire network.

More recent versions of the Explosive trojan contain a configurable option for USB infection. When this option is enabled, Explosive infects any writable mass storage device connected to the server. This can be used to infect additional servers in environments where operational mass storage devices are shared between servers, as well as infect an administrator's home or office machines.

CAMPAIGN TARGETS

Our investigation has revealed a large number of the campaign targets. This was done by means of sinkholing large parts of the C&C infrastructure as well as open-source intelligence (OSINT) efforts and communication we have established with infected companies and organizations.

We identified targets in a wide geographical spread, however, the majority of targets were located in the Middle East. Additionally, for those targets that were not located in the Middle East, we found direct evidence of them having connections to Middle Eastern-based companies and organizations.

According to the specific nature of the targets being attacked, with the majority of targets being from the educational, telecommunication, hosting and civil services sectors, it seems the campaign's overall goal was to attack the countries hosting these targets rather than the individual targets themselves.

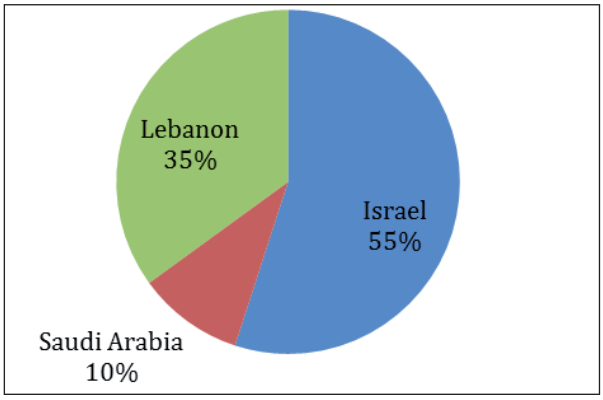

Figure 3: Attacked countries.

TECHNICAL ANALYSIS

The Explosive trojan contains two major components: the main executable binary and a DLL file containing 'backend' API calls.

The main executable file contains most of the trojan logic, while the DLL primarily contains exported actionable API functions. The Explosive DLL file is dynamically loaded by the main executable at runtime whenever it is needed, and unloaded when the desired action is complete.

This separation is probably designed to support quick functionality patches by the attackers, and to avoid heuristic detection of the main executable by common AV engines and other protection software (Table 1).

Both the main executable and the DLL are compiled as a standard VC++ application. The main executable is a console application which supports several optional command-line arguments used to control the trojan's behaviour (Table 2).

Exported DLL function	Description	Version
CON	Main communication API.	All
GetAllData	Collect extensive data from user, OS and applications.	All
GetIEHistory	Get *Internet Explorer*'s history of browsing data.	All
OpenClipFn	OpenClipboard wrapper.	3
PathProcess	Locate and kill currently loaded Explosive modules.	All
SetWinHoK	Wrapper around SetWindowsHookExA.	All
Registerapp	Write Explosive registry values.	All
CreateNewFile	Create a new Explosive instance on external mass storage device.	1, 2
Fdown	URLDownloadToFile wrapper.	1

Table 1: Exported DLL functions.

Option	Function
-i	Install the Explosive trojan as a service. The service is usually created with a blank description.
-h \ -x	Force the Explosive trojan to initiate a 20-second delay on startup.
-d	Stop the Explosive process, and remove all traces of infection from the system.

Table 2: Command-line arguments used to control the trojan's behaviour.

Thread #	Description
Key Logger	A basic implementation of a *Windows* key logger using the SetWindowsHookEx API call.
Clipboard Logger	Logs all clipboard data implemented by periodically opening and peeking into the current user clipboard data.
Memory Monitor	Constantly monitors Explosive's memory consumption by calling the GetProcessMemoryInfo API and reading WorkingSetSize.
C&C Secure Checks	Periodically connects to the C&C server with a special connection string, and determines if the connection is secure by the return of a predefined value. If the connection is not secure, all operations are stopped until a secure connection is achieved.

Table 3: Explosive's threads.

Once installed, the Explosive trojan creates several threads to support its functionality (Table 3).

EXPLOSIVE VERSIONS

Over the entire attack timeline, we detected five different versions of Explosive, as shown in Table 4.

Explosive version	Description
Version 1	Un-obfuscated network traffic.
Version 2	Most common version, clipboard monitoring added.
Version 3	Most advanced version detected.
KS version	Uses only keyboard and clipboard hooking modules.
Micro	Possible ancestor. Uses the same C&C server framework.

Table 4: Explosive versions.

Version 1 is the earliest version of Explosive, with the first compiled sample dated to November 2012. This version includes very basic backdoor features, and C&C communication is not obfuscated. The default C&C server is no longer active, and we believe that no infections of this version are currently active.

Versions 2 and 3 are more mature implementations of the Explosive trojan, with added concealment and operational features as well as a new set of supported actions for C&C commands.

The KS version is very similar in functionality to other Explosive versions. However, this version has no communication functionality and is probably used by the attackers to avoid network detection in special cases. This version stores the extracted server data on the server's file system to be downloaded later by the attacker using the pre-installed web shell.

Micro seems to be an early ancestor of the Explosive trojan. Only a few samples of it were detected. Micro does not use the same C&C server or protocol as the other versions, but uses the 'dynamic updater' framework to pass commands via HTTP.

CONFIGURATION

Each of the main Explosive binary files contains an integrated configuration section, which is located at a fixed position in the binary image overlay. The configuration section itself is not encrypted but the readable configuration values are stored as obfuscated strings.

As expected, the configuration section has evolved with subsequent versions of Explosive, and newer versions present new configuration parameters.

OBFUSCATION

Explosive uses custom obfuscation techniques to encode configuration values, C&C communication and C&C updating protocols. The obfuscation algorithm is not very advanced and does not attempt to merge the obfuscated data into its surroundings. The primary motivation for this obfuscation appears to be to avoid detection by automated security tools such as anti-virus or IPS engines.

Configuration encoding

Both the configuration and C&C updating data use a custom ASCII encoding algorithm in which each plaintext character is transformed into its hex ASCII value equivalent and separated by a '@' sign (see Figure 4).

Communication encoding

Starting from version 2, C&C network traffic is encoded using a custom algorithm. To encode the data, the plaintext bytecode is reversed, base64 encoded, and reversed again (see Figure 5).

COMMUNICATION

Explosive's communication algorithm is very complex and contains many, often unnecessary, branches and loops.

A hard-coded C&C IP address in embedded in Explosive's main module. Explosive initially attempts to connect to this preset C&C address. If the C&C server is non responsive, the hard-coded static updater server is contacted to obtain an updated C&C address. If the static updater is also non responsive, a custom DGA algorithm is used to produce a 'dynamic updater' domain name, which is a secondary C&C updater server. This server has the same functionality as the static server, with the exception of its operating URI. Our investigation led us to two values used as DGA seeds: 'redotntexplore' and 'flashplayergetadobe'.

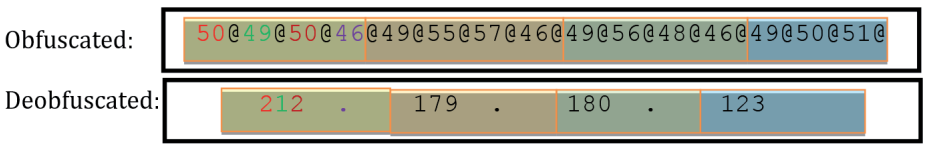

Figure 4: Configuration encoding.

Figure 5: Communication encoding.

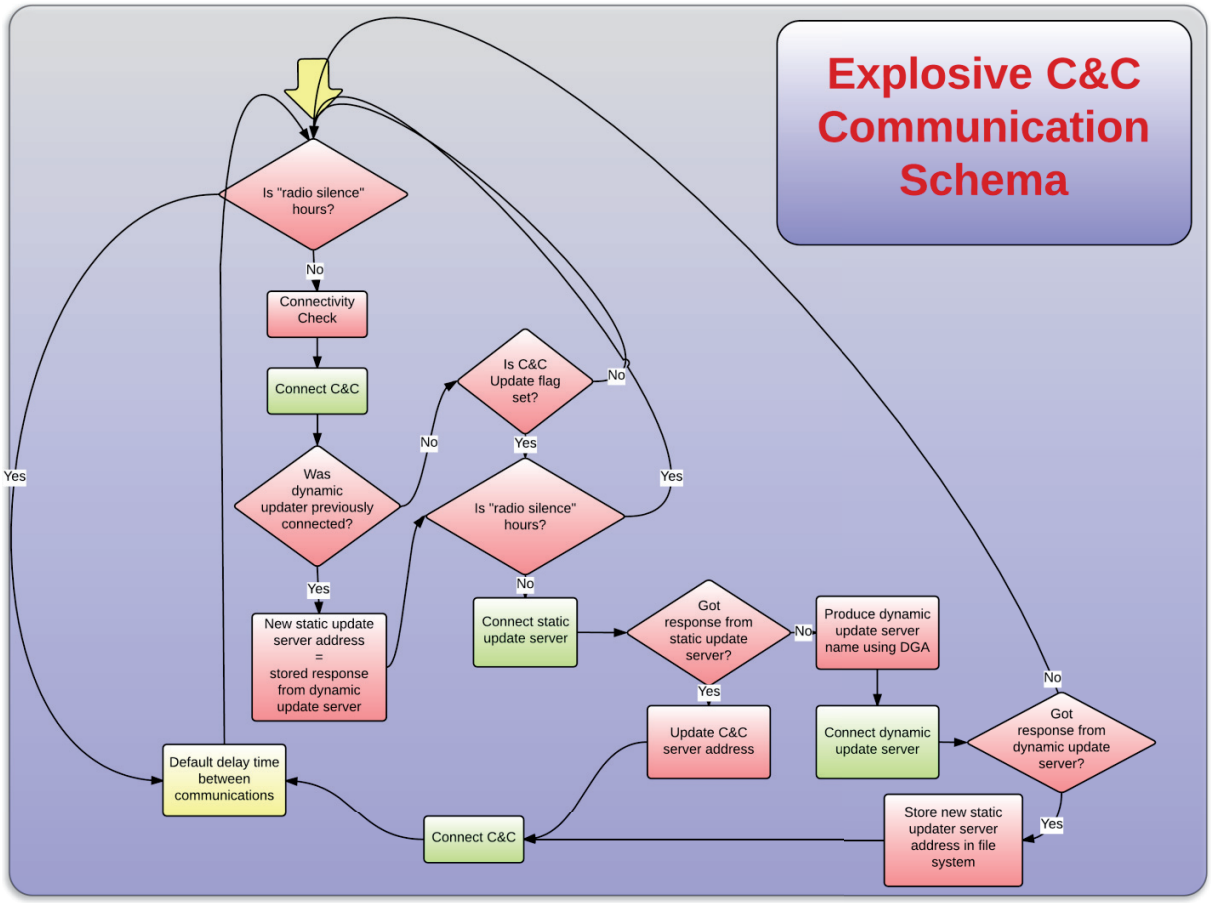

Figure 6: Explosive C&C communication.

Parameter	Description
Password	A fixed (encoded) password field. This value remains the same for all analysed Explosive versions. The decoded password value is: <*'1Q2W3E4r1'*>
Identifier	A value identifying the specific Explosive version and port.
Client External IP	The IP of the gateway connecting this IP to the Internet. This value is extracted from a query to 'whatismyip2.somee.com' or 'api.externalip.net' that takes place just before the initial C&C communication. If both of the 'what-is-my-ip' services are unavailable, a custom service with similar functionality located at the C&C server over TCP/8084 is connected. If all queries fail, this value is set to '0.0.0.0' (or local IP in some versions).
Username\PID	The current logged in username and process ID.
Hostname	The infected host name.
System Name	The running OS, retrieved from the 'systeminfo' CLI command output.
Installation path	Full path and file name of the current executable.

Table 5: Information sent during initial C&C communication.

C&C communication

The C&C communication is performed using raw TCP sockets and encoded (with the exception of Explosive version 1, which does not encode its C&C traffic) using the previously mentioned communication encoding scheme.

Once the Explosive module successfully initiates communication with its C&C server, it sends an authentication password and additional data identifying the infected target (Figure 7).

Table 5 shows the information contained in the initial C&C request.

Next, the C&C server responds with a confirmation message, followed by an optional list of commands for the Explosive module. The confirmation message always starts with the encoded string '<!*connectok*!>'.

```
[*] C&C Request Received
    Password: ==gKg5XI+BmK8oCYxEFQXNSR0IXMgpiP
    Identifier: Explosive-443<TrVs>v3</TrVs>
    Client internal IP: 2.3.4.5
    User Name: Administrator:1334
    System Name: RESEARCH
    Host Name: Windows XP
    Installation Path: c:\windows\system\evil.exe
```

Figure 7: Explosive sends an authentication password and additional data identifying the infected target.

Decoded C&C command	Description
DumpHist	Dump *IE* history.
DumpPass	Dump saved passwords.
GetRegValue	Get a specified registry value.
ListProcess	List all running processes.
RunCmd	Run a specified command line.
GetFile	Send a specific file to the C&C server.
UnZip<	Decompress a specified file to folder.
DeleteFiles<	Delete specified files.
GetDrivesFolder<	Get the content of a specific folder.
<!*KILL*!>	Kill Explosive process.
<!*RERUN*!>	Restart Explosive process.
<!*DEL*!>	Kill Explosive process and remove all traces.

Table 6: Subset of Explosive C&C commands and their description.

Table 6 shows a subset of Explosive C&C commands and their description.

As both the Explosive C&C requests and responses use raw TCP sockets and start with the same static 'message delimiter' parameter, traffic containing the TCP payload starting with the string '==gKg5XI+BmK' can be used as a network indicator for Explosive C&C communication.

Static/dynamic updaters

The static updater is installed on a single web server, and its URL is hard coded as part of the Explosive configuration section.

To disguise the server, its default (root) web page is a ripped HTML page from a random Internet site with all links and functionality redirecting to the original site.

Once the Explosive client generates a GET request to a specific URI, a custom HTTP response is returned with a unique identifier, and the IP address and port of the new C&C server.

As opposed to the static updater, the dynamic updater does not contain a hard-coded address value in the configuration section. Instead, it uses an initial value as an input argument for a custom DGA algorithm to produce the server address.

The same routine used by the static updater for updating the C&C data is used on each DGA algorithm result until a verified answer is received. Once this occurs, the DGA algorithm terminates and the current updater is set as the new static updater server.

The resulting address from the DGA algorithm can be one of 170 possible permutations of the initial value.

Several indicators can be used to identify all Explosive HTTP communications:

1. The same user agent value is used in all HTTP requests. This user agent is hard coded into the Explosive DLL binaries, and does not seem to be used by any legitimate application.

```
"Mozilla/4.0 (compatible; MSIE 7.0; MSIE 6.0;
Windows NT 5.1; .NET CLR 2.0.50727)"
```

2. All GET requests are made to a URI starting with an uncommon double slash value.

```
"GET //v2/443/index.php?win=4"
```

Connectivity checks

Connectivity checks are made at several stages of the malware communication algorithm. Explosive attempts to connect to several well-known sites to verify that the infected host is connected to the Internet. For reasons not yet fully understood, the results of these checks are completely disregarded, and the communication algorithm continues normally regardless.

The list of sites checked for connectivity is slightly different in different versions of Explosive. The latest version contains the following sites:

- microsoft.com
- maktoob.yahoo.com
- bing.com
- google.com

An interesting fact to note is that maktoob.yahoo.com is the main *Yahoo* site for Arabic language readers.

CONCLUSIONS

A technical analysis of the Volatile Cedar campaign, as well as the background data collected during the campaign investigation process, has led us to believe that the actor behind this campaign is a well-managed and well-funded organization with no direct financial gains associated with the campaign.

It is also obvious that this organization has a clear interest in attacking Middle Eastern-based organizations, which probably aligns with its political agenda.

By following the trail of evidence collected throughout the campaign investigation, we feel that we have sufficient evidence to conclude that the actor behind this campaign is a Lebanese organization with clear political motives.

If our assumption is correct, this marks a major shift in the APT landscape, showing us that APT campaigns are no longer the sole preserve of very large, multimillion-dollar-budget intelligence agencies. It teaches us that any organization that is sufficiently dedicated to the cause and that has sufficient technological abilities may run a very successful and undetectable APT campaign for a long period of time.