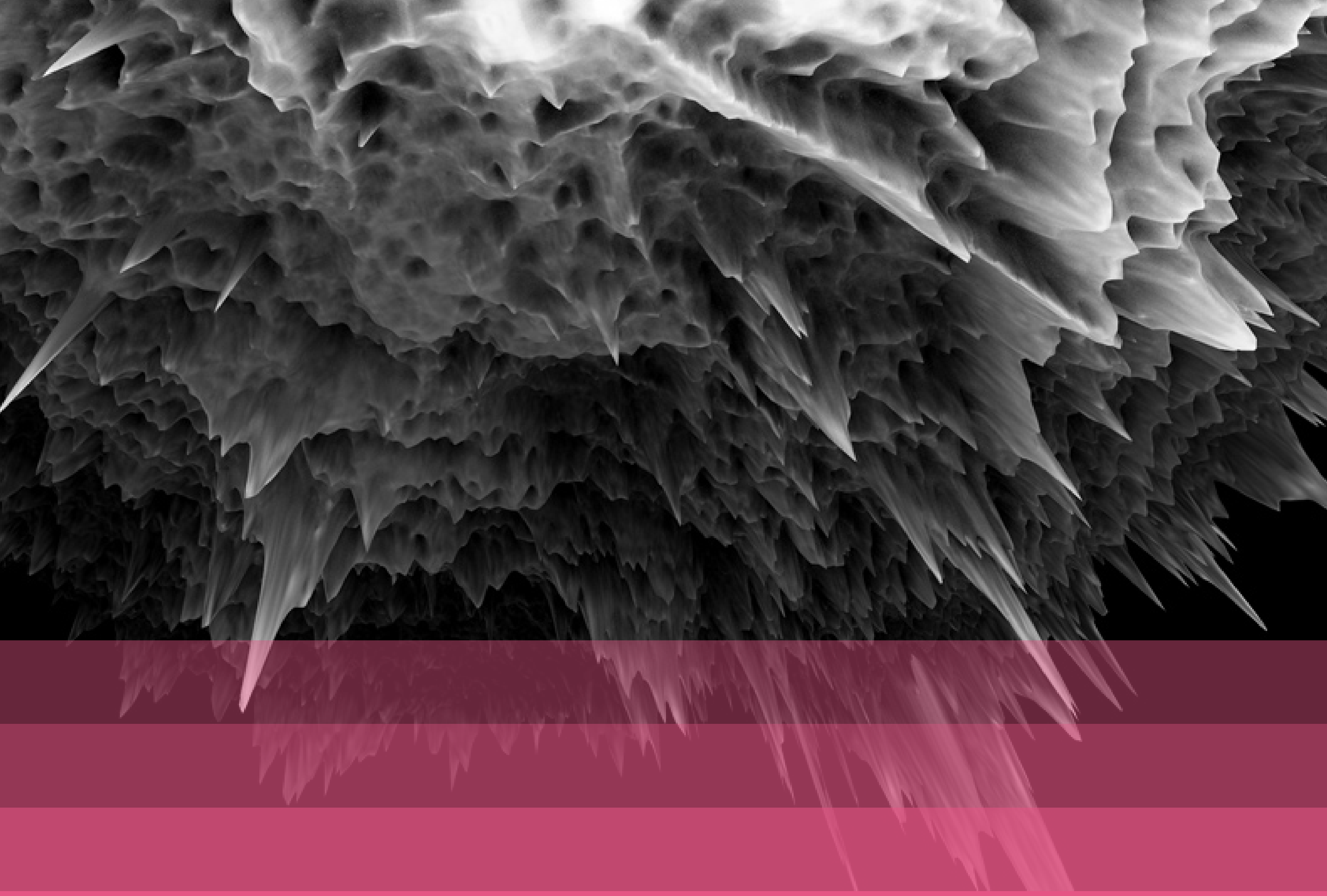


VOLATILE CEDAR

THREAT INTELLIGENCE AND RESEARCH

MARCH 30, 2015

Check Point®
SOFTWARE TECHNOLOGIES LTD.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Beginning in late 2012, a carefully orchestrated attack campaign we call *Volatile Cedar* has been targeting individuals, companies and institutions worldwide. This campaign, led by a persistent attacker group, has successfully penetrated a large number of targets using various attack techniques, and specifically, a custom-made malware implant codenamed *Explosive*.

This report provides an extended technical analysis of *Volatile Cedar* and the *Explosive* malware.

Malware attribution is often tricky and deception-prone. With that in mind, investigation of the evidence leads us to suspect *Volatile Cedar* originates from Lebanon (hence its nickname). Moreover, the *Volatile Cedar* target vertical distribution strongly aligns with nation-state/political-group interests, eliminating the possibility of financially motivated attackers.

We have seen clear evidence that *Volatile Cedar* has been active for almost 3 years. While many of the technical aspects of the threat are not considered "cutting edge", the campaign has been continually and successfully operational 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 rapidly responds to detection incidents.

Volatile Cedar is heavily based on a custom-made remote access Trojan named *Explosive*, which is implanted within its targets and then used to harvest information. Tracking down these infections was quite a difficult task due to the multiple concealment measures taken by the attackers. The attackers select only a handful of targets to avoid unnecessary exposure. New and custom versions are developed, compiled and deployed specifically for certain targets, and "radio silence" periods are configured and embedded specifically into each targeted implant.

The modus operandi for this attacker group initially targets publicly facing web servers, with both automatic and manual vulnerability discovery. Once in control of a server, the attackers further penetrate the targeted internal network via various means, including manual online hacking as well as an automated USB infection mechanism.

We will discuss the attack vectors and infection techniques used by the attack campaign as well as provide indicators that can be used to detect and remove the infection.

For hashes, domains, IP addresses and other indicators of compromise, see **Appendix C**.

Some of the details in this investigation were edited or omitted from this report to protect customer privacy and ongoing research efforts. Further information may be released in future reports.

OVERVIEW

Volatile Cedar is a highly targeted and very well-managed campaign. Its targets are carefully chosen, confining the infection spread to the bare minimum required to achieve the attacker's 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 servers serve as publicly exposed, easily accessible gateways to private and more secure internal networks. As these servers have a common business functionality, their security is often sacrificed for productivity, making them an easy target for attackers. Once the 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.

The 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 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 any *Explosive* version 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 this publication. See the figure below for more details.

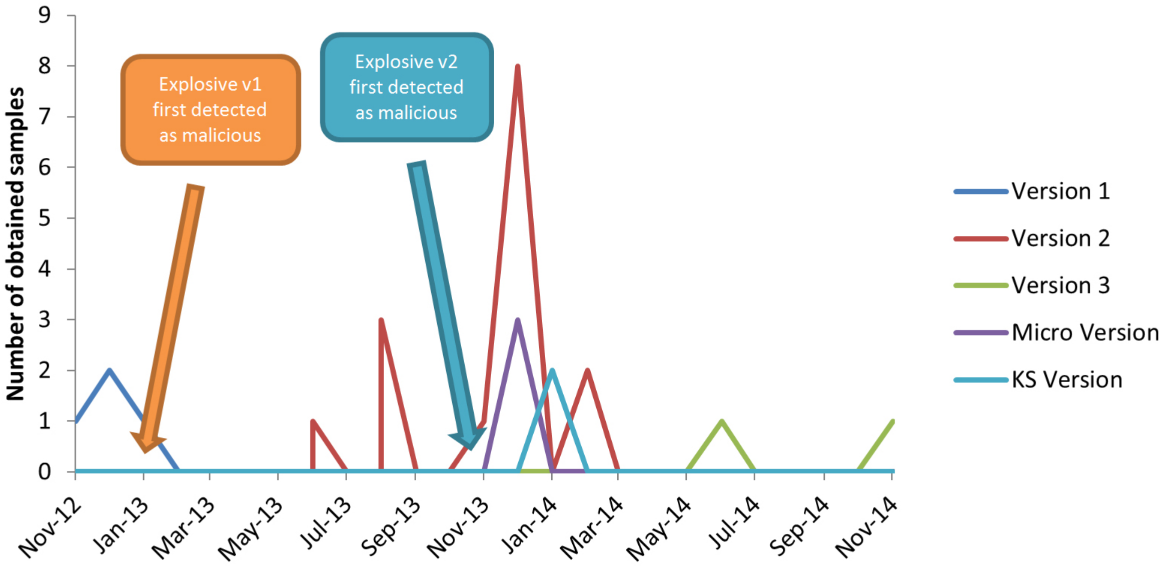


Figure 1 - Explosive version timeline

STEALTH

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

- AV detections are avoided by frequently checking AV results and changing versions and builds on all infected servers when any traces of detection appear. See Figure 1.
- New versions 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.
- Obfuscated C&C communication may appear as random network traffic "noise" to certain network inspection devices.
- 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.

CONTROL NETWORK

The campaign uses a multi-tiered server backend framework to control the targeted systems. This backend framework is composed of 3 major tiers:

- **Tier 1—C&C servers**: Each *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 hardcoded 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 hardcoded as part of the *Explosive* configuration section.
- **Tier 3—Dynamic update servers:** If the static C&C server is nonresponsive, the *Explosive* infection initiates a custom DGA algorithm which attempts to connect to the dynamic update servers. Once connected, these servers operate 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, other servers use publicly shared hosting frameworks or even compromised legitimate servers.

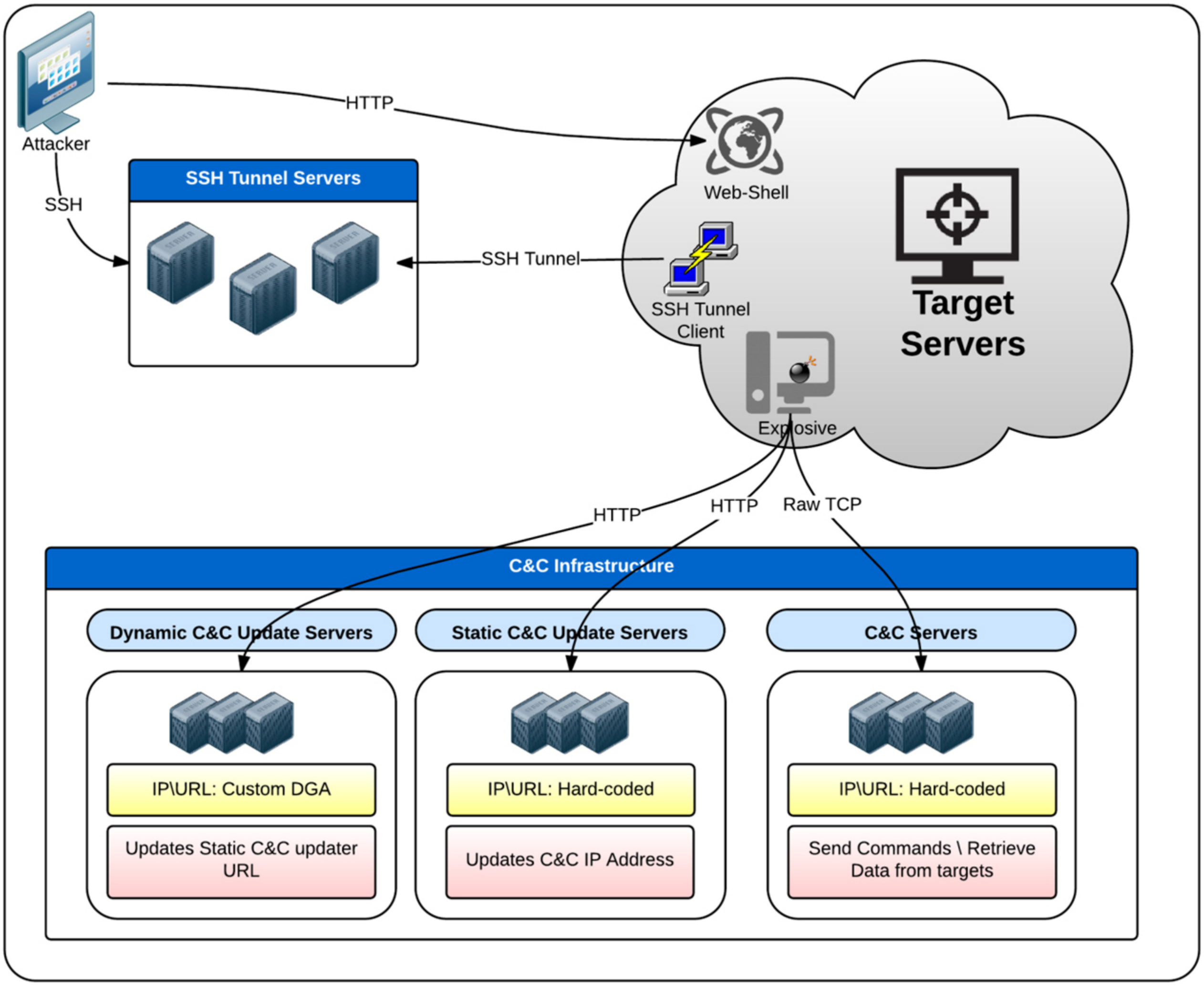


Table 1 - Explosive server infrastructure

INFECTION SPREAD

Evidence shows that the *Explosive* Trojan leverages its 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 manually spread 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.
For additional information on the USB infection process, see **Appendix B.**

ATTRIBUTION

Malware attribution has always been a difficult task and *Volatile Cedar* is no different. Although we have no hard evidence upon which to base our conclusions, and many of the factors we rely on can in theory be forged or misinterpreted, we believe the unique combination of these factors reveal the attacker's agenda and provide a good estimation of his whereabouts.

1. To assign a rough geographical location, we observed the UTC creation times of detected samples. The results can be seen in the following table:

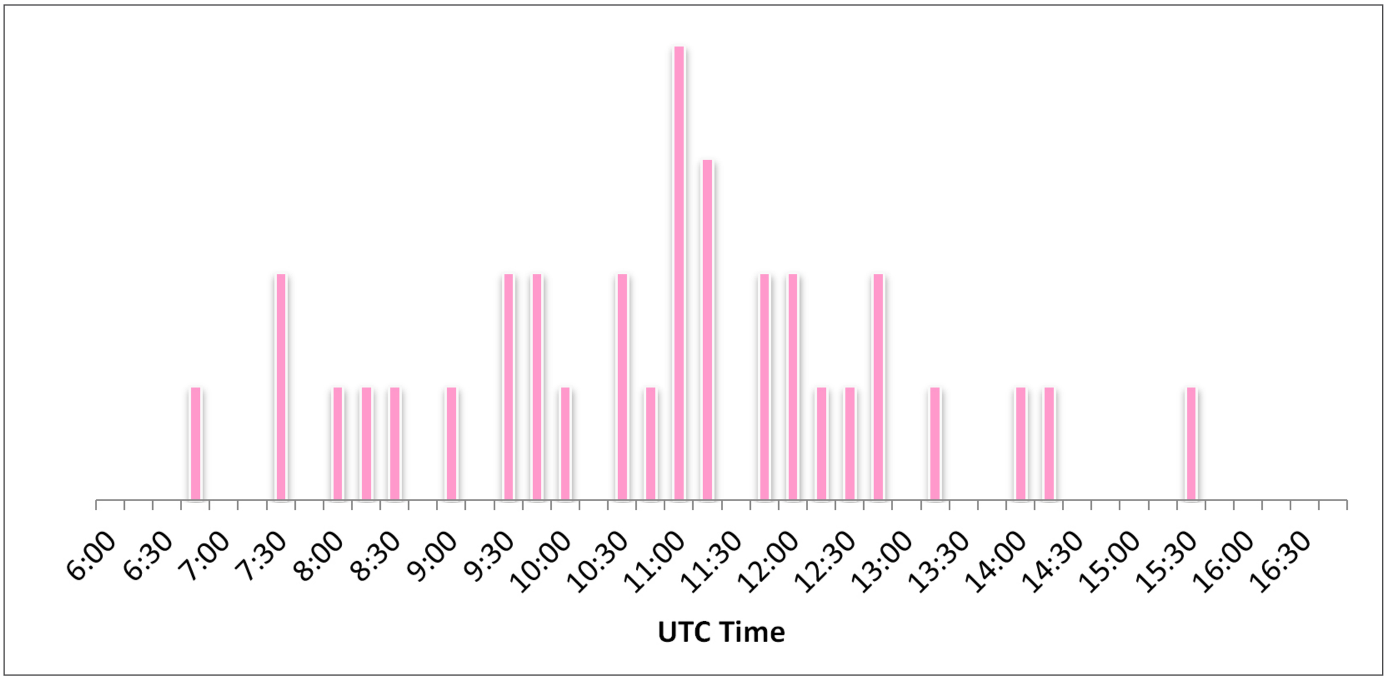


Figure 2 - Explosive sample UTC compilation hours

With conventional working hours usually between 08:00-17:00, the creation locale time can be comfortably correlated to GMT+2.

2. To further align our results, we took into account several other factors collected from the C&C server infrastructure:

- The C&C servers for the first *Explosive* version were hosted at a major Lebanese hosting company. This is not commonly seen in the malware arena.
- DNS registrant information from several of the infrastructure servers shows that they are or were previously registered under contacts with a very similar Lebanese address.
- Careful observation of DNS registrant contact information history has revealed an OPSEC failure by the attackers in one instance. For a brief period (possibly before the server was operational), WHOIS privacy was inactive, pointing at a real identity of the registrant. This e-mail address leads to social media accounts that show public and clear affinity with Lebanese political activism.

While not all of the targets have been identified yet, we can start building a profile of the intended victims. Some of the confirmed targets can be associated with organizations related to the state of Israel, and some are Lebanon-based, potentially testifying to in-state espionage among rival political groups.

Other factors to consider are the low infection rate and the targeted nature of this campaign. These suggest that the attacker's motives are not financial but aim to extract sensitive information from the targets.The combination of these factors leads us to believe that the attack originated or is sponsored by groups affiliated with Lebanon and the specific targets are chosen based on nation-state/political-group interests.

EXPLOSIVE ANALYSIS

The *Explosive* Trojan contains 2 major components:

- The main executable binary
- 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.

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 2 - Common Explosive DLL functions

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 behavior:

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 a 20 second delay on startup.
-d	Stop the *Explosive* process, and remove all traces of infection from the system.

Table 3 - Optional command line options

Once installed, the *Explosive* Trojan creates several threads to support its functionality:

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 4 - Main explosive threads

EXPLOSIVE VERSIONS

Over the entire attack timeline, we detected 5 different versions of *Explosive*:

***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 5 - Explosive versions

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

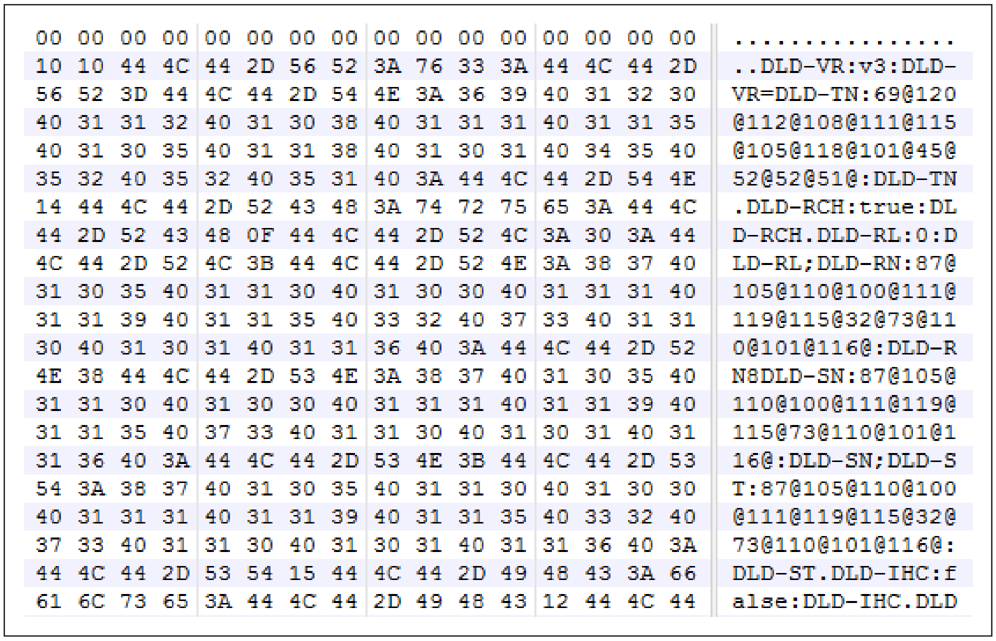
Version 2 and **Version 3** are more mature implementations of the *Explosive* Trojan, with added concealment and operational features as well as a new set of supported actions for C&C commands.

The **KS version** is very similar in functionality to other *Explosive* versions. However, this version has no communication functionality and is most 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. For more details of the Micro version, see **Appendix A**.

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.



```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
10 10 44 4C 44 2D 56 52 3A 76 33 3A 44 4C 44 2D  ..DLD-VR:v3:DLD-
56 52 3D 44 4C 44 2D 54 4E 3A 36 39 40 31 32 30  VR=DLD-TN:69@120
40 31 31 32 40 31 30 38 40 31 31 31 40 31 31 35  @112@108@111@115
40 31 30 35 40 31 31 38 40 31 30 31 40 34 35 40  @105@118@101@45@
35 32 40 35 32 40 35 31 40 3A 44 4C 44 2D 54 4E  52@52@51@:DLD-TN
14 44 4C 44 2D 52 43 48 3A 74 72 75 65 3A 44 4C  .DLD-RCH:true:DL
44 2D 52 43 48 0F 44 4C 44 2D 52 4C 3A 30 3A 44  D-RCH.DLD-RL:0:D
4C 44 2D 52 4C 3B 44 4C 44 2D 52 4E 3A 38 37 40  LD-RL;DLD-RN:87@
31 30 35 40 31 31 30 40 31 30 30 40 31 31 31 40  105@110@100@111@
31 31 39 40 31 31 35 40 33 32 40 37 33 40 31 31  119@115@32@73@11
30 40 31 30 31 40 31 31 36 40 3A 44 4C 44 2D 52  0@101@116@:DLD-R
4E 38 44 4C 44 2D 53 4E 3A 38 37 40 31 30 35 40  N8DLD-SN:87@105@
31 31 30 40 31 30 30 40 31 31 31 40 31 31 39 40  110@100@111@119@
31 31 35 40 37 33 40 31 31 30 40 31 30 31 40 31  115@73@110@101@1
31 36 40 3A 44 4C 44 2D 53 4E 3B 44 4C 44 2D 53  16@:DLD-SN;DLD-S
54 3A 38 37 40 31 30 35 40 31 31 30 40 31 30 30  T:87@105@110@100
40 31 31 31 40 31 31 39 40 31 31 35 40 33 32 40  @111@119@115@32@
37 33 40 31 31 30 40 31 30 31 40 31 31 36 40 3A  73@110@101@116@:
44 4C 44 2D 53 54 15 44 4C 44 2D 49 48 43 3A 66  DLD-ST.DLD-IHC:f
61 6C 73 65 3A 44 4C 44 2D 49 48 43 12 44 4C 44  alse:DLD-IHC.DLD
```

Figure 3 - Explosive Configuration Section

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

Parameter Name	Description	Version
DLD-ACT	*Explosive* constantly attempts to update its C&C IP address when this flag is set.	All
DLD-C	A unique identifier used for updating C&C communication.	All
DLD-C0	Same as DLD-C.	All
DLD-D	URL for the static C&C updater.	All
DLD-E	TLD of the dynamic C&C updater.	All
DLD-P	Path for the dynamic C&C updater.	All
DLD-IHC	No communication is generated during "silent mode" when this flag is set.	2, 3
DLD-IH1	Starting hour of "silent mode."	2, 3
DLD-IH2	Ending hour of "silent mode."	2, 3
DLD-PRT	Default C&C Port.	All
DLD-IP	Default C&C IP address.	All
DLD-OIP	Other (additional) C&C IP addresses.	3
DLD-NTI	Delay time between C&C connections.	All
DLD-RCH	Registration related.	2, 3
DLD-RL	Registration related.	2, 3
DLD-RN	Registry key name.	All
DLD-S	Initial value for dynamic C&C updating DGA.	All
DLD-SN	Installed service name.	2, 3
DLD-ST	Installed service type.	2, 3
DLD-TN	Unique identifier for C&C communication.	All
DLD-USA	Removable device infection method.	All
DLD-USI	Removable device infection flag.	All

Table 6 - Configuration parameters

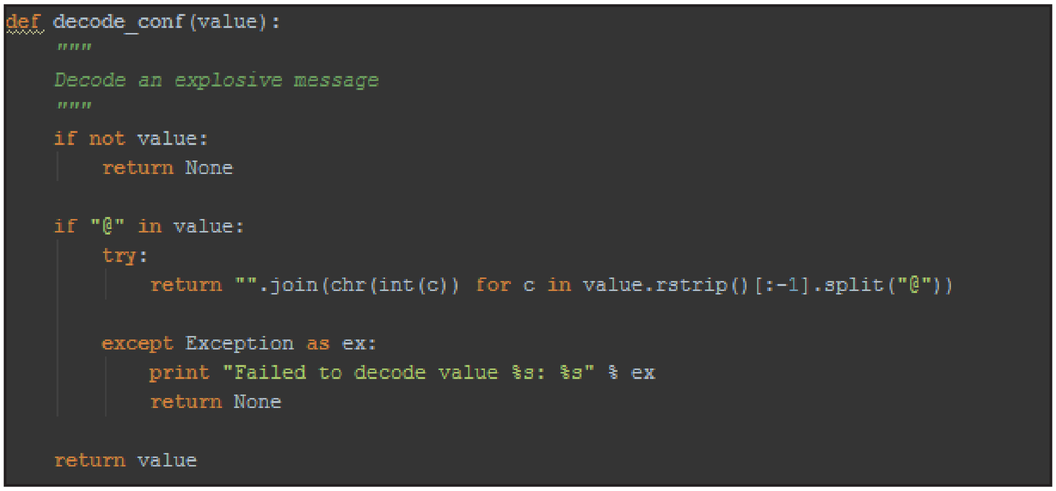
OBFUSCATION

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 antivirus 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.
For example, the configuration value:
`"50@49@50@46@49@55@57@46@49@56@48@46@49@50@51@"`
is decoded into the plaintext string:
`"212.179.180.123".`

The following Python code can be used to encode\decode the configuration strings:



```
def decode_conf(value):
    """
    Decode an explosive message
    """
    if not value:
        return None

    if "@" in value:
        try:
            return "".join(chr(int(c)) for c in value.rstrip()[:-1].split("@"))

        except Exception as ex:
            print "Failed to decode value %s: %s" % ex
            return None

    return value
```

Figure 4 - Configuration parameter decoding

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.

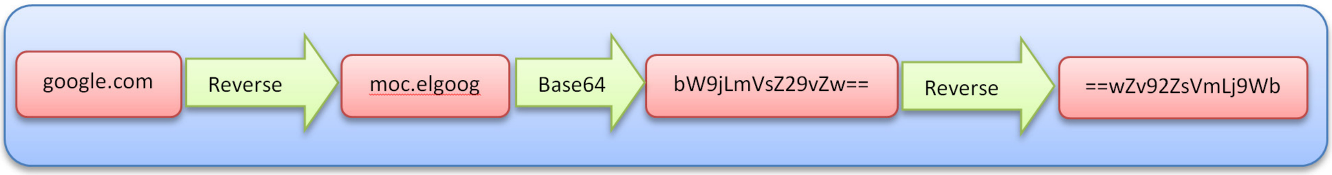


Figure 5 - Communication encoding scheme

COMMUNICATION

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

A hardcoded 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 nonresponsive, the hardcoded static updater server is contacted to obtain an updated C&C address. If the static updater is also nonresponsive, 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.

The only DGA initial value we observed in our obtained samples was 'redotntexplore'.

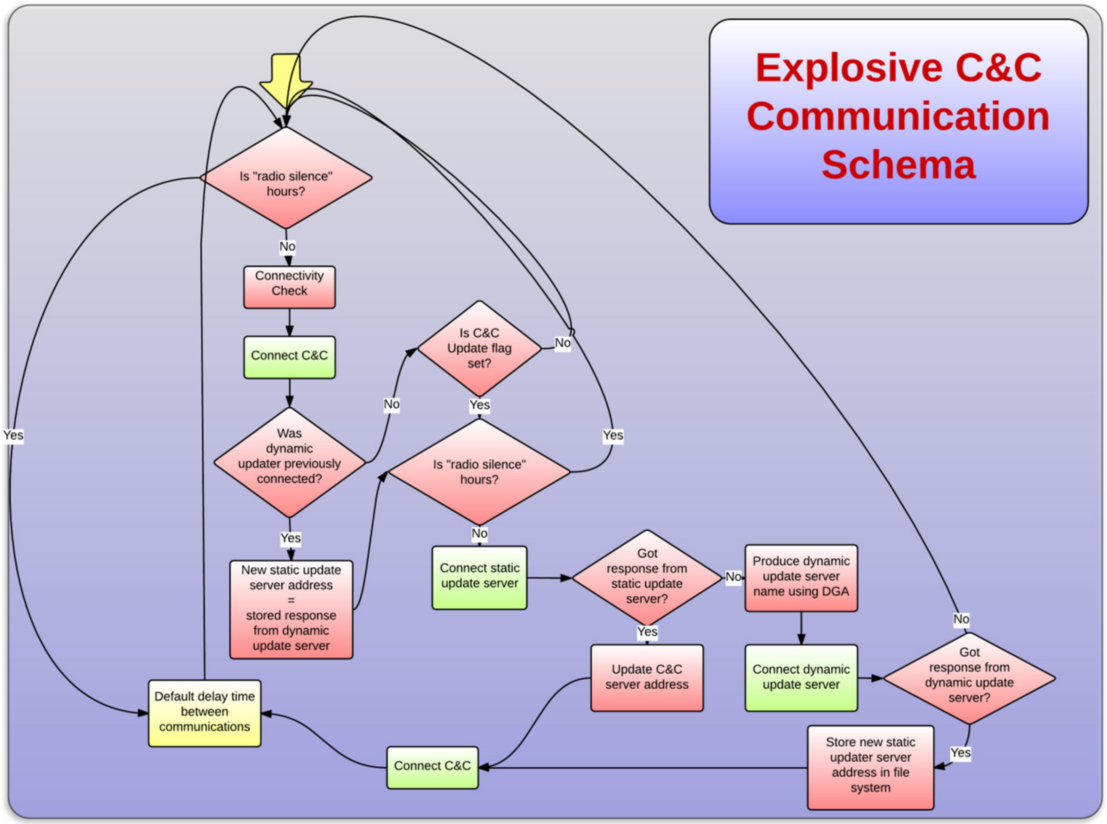
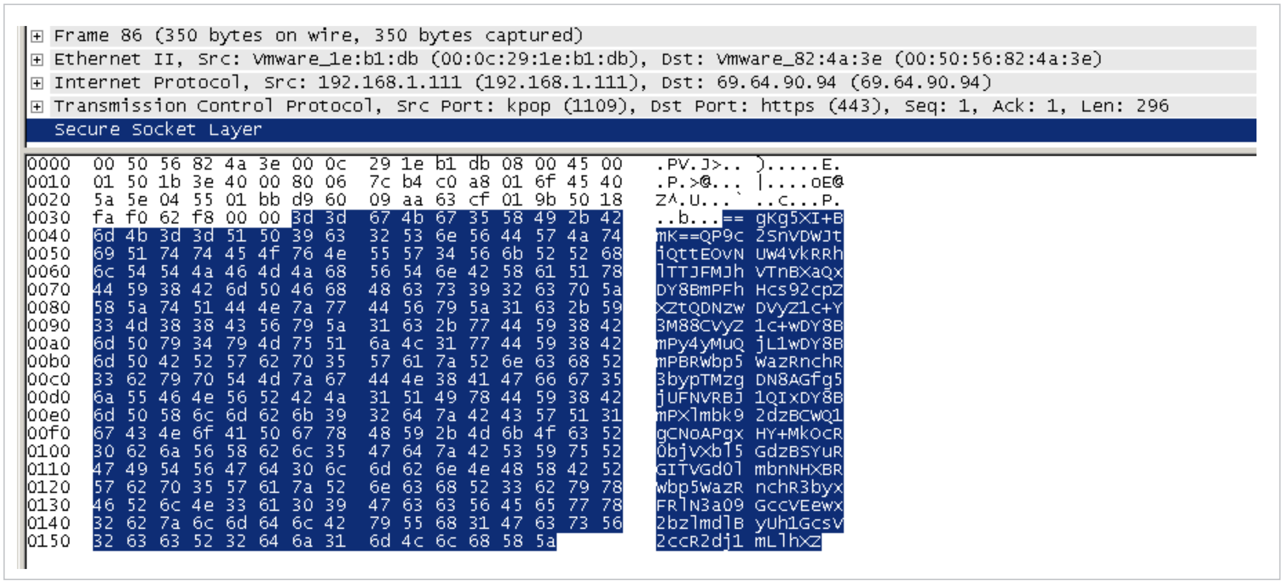

Explosive C&C Communication Schema
Is "radio silence" hours?
No
Connectivity Check
Connect C&C
Was dynamic updater previously connected?
Yes
New static update server address = stored response from dynamic update server
No
Is C&C Update flag set?
No
Yes
Is "radio silence" hours?
Yes
No
Connect static update server
Got response from static update server?
No
Produce dynamic update server name using DGA
Yes
Update C&C server address
Connect dynamic update server
No
Got response from dynamic update server?
Yes
Store new static updater server address in file system
Connect C&C
Default delay time between communications
Yes

Figure 6 - Explosive C&C Communication Schema

C&C COMMUNICATION

The C&C communication is performed using raw TCP sockets and encoded[1] using the previously mentioned communication encoding scheme.

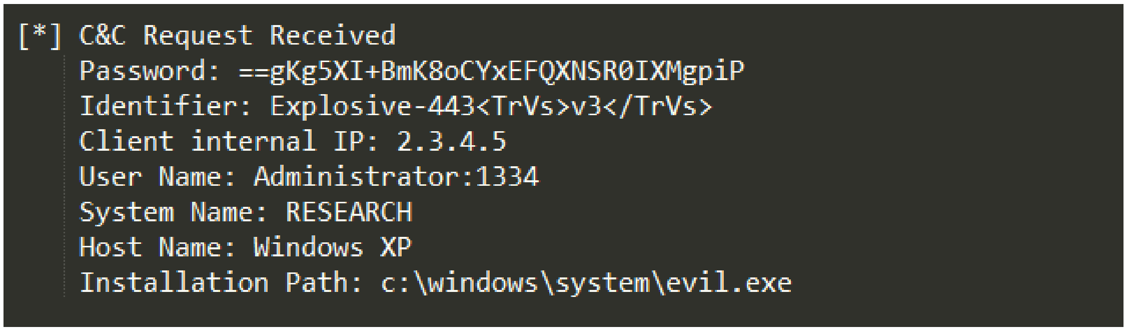
Once the *Explosive* module successfully initiates communication with its C&C server, it sends an authentication password and additional data identifying the infected target.



```
⊞ Frame 86 (350 bytes on wire, 350 bytes captured)
⊞ Ethernet II, Src: Vmware_1e:b1:db (00:0c:29:1e:b1:db), Dst: Vmware_82:4a:3e (00:50:56:82:4a:3e)
⊞ Internet Protocol, Src: 192.168.1.111 (192.168.1.111), Dst: 69.64.90.94 (69.64.90.94)
⊞ Transmission Control Protocol, Src Port: kpop (1109), Dst Port: https (443), Seq: 1, Ack: 1, Len: 296
  Secure Socket Layer

0000  00 50 56 82 4a 3e 00 0c  29 1e b1 db 08 00 45 00   .PV.J>.. ).....E.
0010  01 50 1b 3e 40 00 80 06  7c b4 c0 a8 01 6f 45 40   .P.>@... |....oE@
0020  5a 5e 04 55 01 bb d9 60  09 aa 63 cf 01 9b 50 18   Z^.U...` ..c...P.
0030  fa f0 62 f8 00 00 3d 3d  67 4b 67 35 58 49 2b 42   ..b...== gKg5XI+B
0040  6d 4b 3d 3d 51 50 39 63  32 53 6e 56 44 57 4a 74   mK==QP9c 2SnVDWJt
0050  69 51 74 74 45 4f 76 4e  55 57 34 56 6b 52 52 68   iQttEOvN UW4VkRRh
0060  6c 54 54 4a 46 4d 4a 68  56 54 6e 42 58 61 51 78   lTTJFMJh VTnBXaQx
0070  44 59 38 42 6d 50 46 68  48 63 73 39 32 63 70 5a   DY8BmPFh Hcs92cpZ
0080  58 5a 74 51 44 4e 7a 77  44 56 79 5a 31 63 2b 59   XZtQDNzw DVyZ1c+Y
0090  33 4d 38 38 43 56 79 5a  31 63 2b 77 44 59 38 42   3M88CVyZ 1c+wDY8B
00a0  6d 50 79 34 79 4d 75 51  6a 4c 31 77 44 59 38 42   mPy4yMuQ jL1wDY8B
00b0  6d 50 42 52 57 62 70 35  57 61 7a 52 6e 63 68 52   mPBRWbp5 WazRnchR
00c0  33 62 79 70 54 4d 7a 67  44 4e 38 41 47 66 67 35   3bypTMzg DN8AGfg5
00d0  6a 55 46 4e 56 52 42 4a  31 51 49 78 44 59 38 42   jUFNVRBJ 1QIxDY8B
00e0  6d 50 58 6c 6d 62 6b 39  32 64 7a 42 43 57 51 31   mPXlmbk9 2dzBCWQ1
00f0  67 43 4e 6f 41 50 67 78  48 59 2b 4d 6b 4f 63 52   gCNoAPgx HY+MkOcR
0100  30 62 6a 56 58 62 6c 35  47 64 7a 42 53 59 75 52   0bjVXbl5 GdzBSYuR
0110  47 49 54 56 47 64 30 6c  6d 62 6e 4e 48 58 42 52   GITVGd0l mbnNHXBR
0120  57 62 70 35 57 61 7a 52  6e 63 68 52 33 62 79 78   Wbp5WazR nchR3byx
0130  46 52 6c 4e 33 61 30 39  47 63 63 56 45 65 77 78   FRlN3a09 GccVEewx
0140  32 62 7a 6c 6d 64 6c 42  79 55 68 31 47 63 73 56   2bzlmdlB yUh1GcsV
0150  32 63 63 52 32 64 6a 31  6d 4c 6c 68 58 5a         2ccR2dj1 mLlhXZ
```

Figure 7 - Initial C&C Request (Encoded)

[1] With the exception of *Explosive* version 1 which does not encode its C&C traffic.



```
[*] 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
```

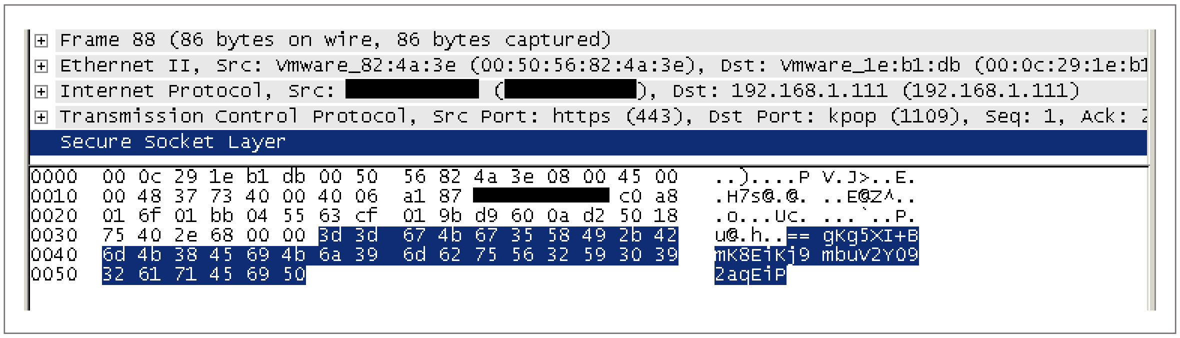
Figure 8 – Initial C&C Request (Decoded)

As seen in Figure 8, the initial C&C request contains the following information:

Parameter	Description
Password	A fixed (encoded) password field. This value remains the same for all analyzed *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 not available, 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	Current executable full path and file name.

Table 7 - Information sent during initial C&C communication

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*!>'.



```
⊞ Frame 88 (86 bytes on wire, 86 bytes captured)
⊞ Ethernet II, Src: Vmware_82:4a:3e (00:50:56:82:4a:3e), Dst: Vmware_1e:b1:db (00:0c:29:1e:b1
⊞ Internet Protocol, Src: [redacted] ([redacted]), Dst: 192.168.1.111 (192.168.1.111)
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: kpop (1109), Seq: 1, Ack:
  Secure Socket Layer

0000  00 0c 29 1e b1 db 00 50  56 82 4a 3e 08 00 45 00   ..)....P V.J>..E.
0010  00 48 37 73 40 00 40 06  a1 87 [redacted] c0 a8   .H7s@.@. ..E@Z^..
0020  01 6f 01 bb 04 55 63 cf  01 9b d9 60 0a d2 50 18   .o...Uc. ...`..P.
0030  75 40 2e 68 00 00 3d 3d  67 4b 67 35 58 49 2b 42   u@.h..== gKg5XI+B
0040  6d 4b 38 45 69 4b 6a 39  6d 62 75 56 32 59 30 39   mK8EiKj9 mbuV2Y09
0050  32 61 71 45 69 50                                  2aqEiP
```

*Figure 9 - C&C <!*connectok*!> Response*

Listed below are a subset of *Explosive* C&C commands and their description (for the complete list, please see **Appendix E**) :

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.

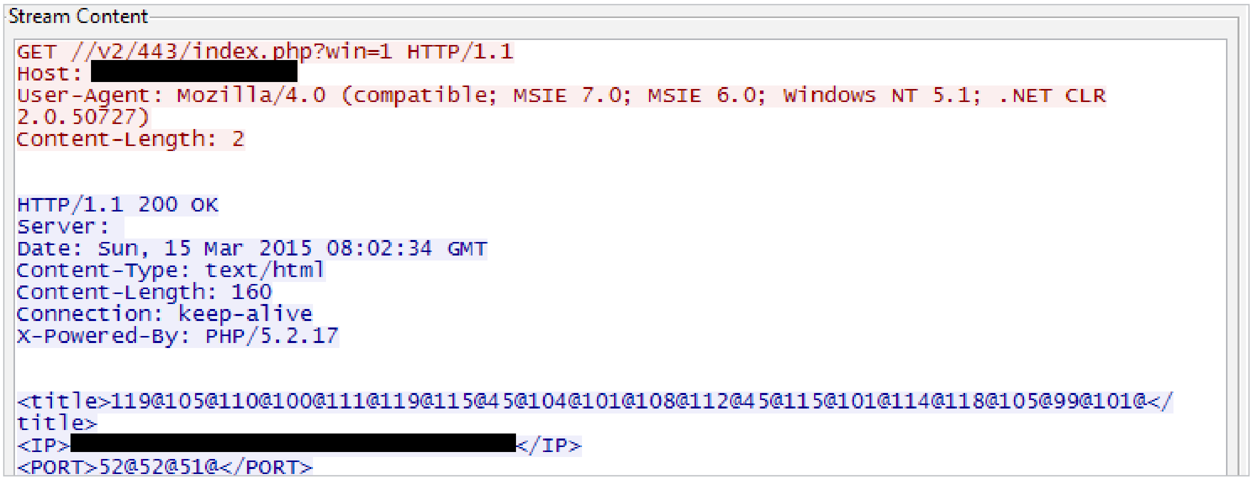
Table 8 – Subset of Explosive C&C commands

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 hardcoded into the *Explosive* configuration section. To disguise the server, the server's 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.



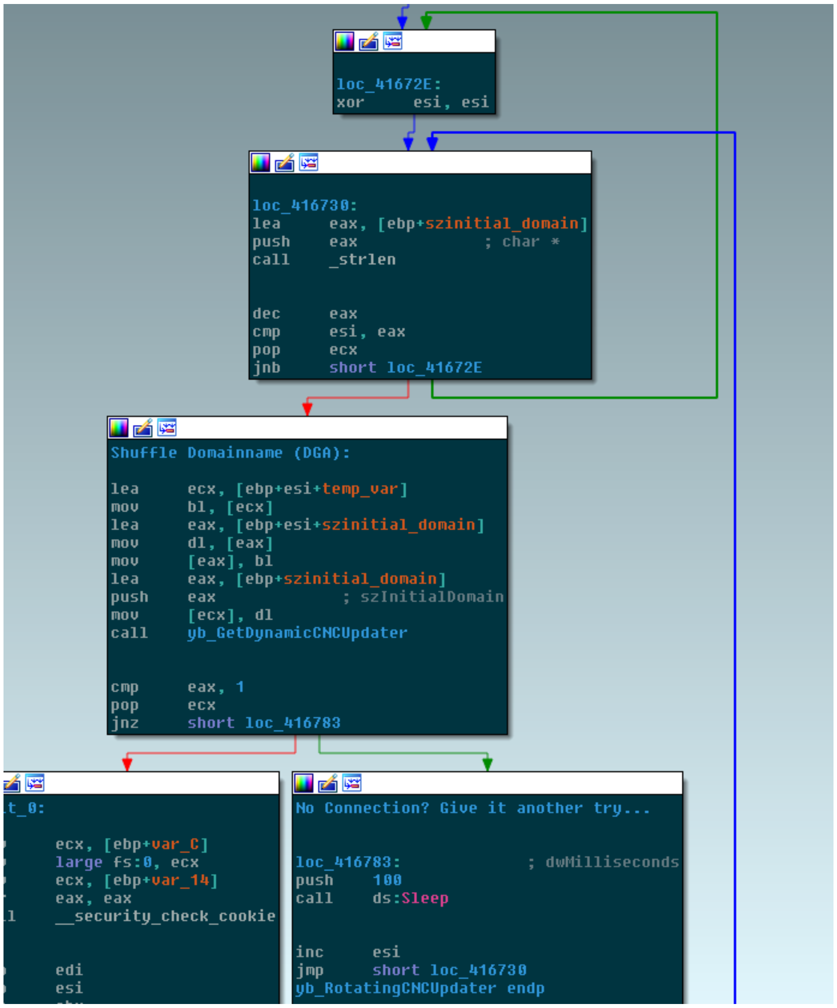
```
GET //v2/443/index.php?win=1 HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; MSIE 6.0; Windows NT 5.1; .NET CLR
2.0.50727)
Content-Length: 2


HTTP/1.1 200 OK
Server:
Date: Sun, 15 Mar 2015 08:02:34 GMT
Content-Type: text/html
Content-Length: 160
Connection: keep-alive
X-Powered-By: PHP/5.2.17


<title>119@105@110@100@111@119@115@45@104@101@108@112@45@115@101@114@118@105@99@101@</
title>
<IP>[redacted]</IP>
<PORT>52@52@51@</PORT>
```

Figure 10 - Static C&C Updater Request

As opposed to the static updater, the **dynamic updater** does not contain a hardcoded 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.



```
loc_41672E:
xor     esi, esi
```

```
loc_416730:
lea     eax, [ebp+szinitial_domain]
push    eax             ; char *
call    _strlen

dec     eax
cmp     esi, eax
pop     ecx
jnb     short loc_41672E
```

```
Shuffle Domainname (DGA):

lea     ecx, [ebp+esi+temp_var]
mov     bl, [ecx]
lea     eax, [ebp+esi+szinitial_domain]
mov     dl, [eax]
mov     [eax], bl
lea     eax, [ebp+szinitial_domain]
push    eax             ; szInitialDomain
mov     [ecx], dl
call    yb_GetDynamicCNCUpdater

cmp     eax, 1
pop     ecx
jnz     short loc_416783
```

```
t_0:

        ecx, [ebp+var_C]
        large fs:0, ecx
        ecx, [ebp+var_14]
        eax, eax
l       __security_check_cookie


        edi
        esi
```

```
No Connection? Give it another try...


loc_416783:             ; dwMilliseconds
push    100
call    ds:Sleep


inc     esi
jmp     short loc_416730
yb_RotatingCNCUpdater endp
```

Figure 11 - Dynamic C&C updater DGA algorithm

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 hardcoded into the *Explosive* DLL binaries, and does not seem to be valid or 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 if 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 various versions of *Explosive*. The latest version contains the following sites:

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

APPENDIX A – MICRO

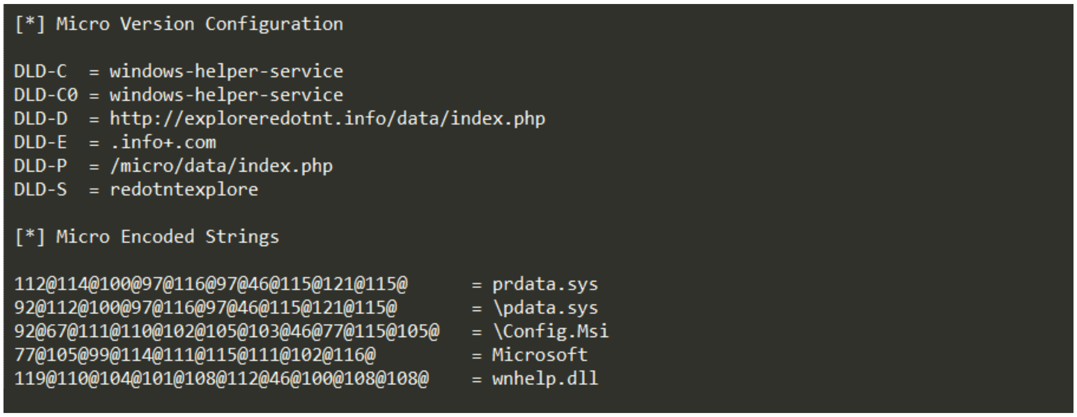
Micro is a rare *Explosive* version. It can best be described as a completely different version of the Trojan, with similarities to the rest of *Explosive* "family" (such as configuration and code base). We believe that Micro is actually an old ancestor of *Explosive*, from which all other versions were developed. As in other versions, this version is also dependent on a self-developed DLL named "wnhelp.dll."

Micro shares the same DGA algorithm as the other versions of *Explosive* and therefore has the same dynamic update server infrastructure. This version, however, uses the dynamic server infrastructure as its C&C server; it connects to a dedicated URI and uses different PHP parameters.

```
http://exloreredotnt.info/micro/data/index.php?micro=4
```

Table 9 – Example of a Micro version URL

Micro has a small configuration (also stored encoded) which uses the same encoding scheme and is located at the binary file overlay.



```
[*] Micro Version Configuration

DLD-C  = windows-helper-service
DLD-C0 = windows-helper-service
DLD-D  = http://exploreredotnt.info/data/index.php
DLD-E  = .info+.com
DLD-P  = /micro/data/index.php
DLD-S  = redotntexplore

[*] Micro Encoded Strings

112@114@100@97@116@97@46@115@121@115@      = prdata.sys
92@112@100@97@116@97@46@115@121@115@       = \pdata.sys
92@67@111@110@102@105@103@46@77@115@105@   = \Config.Msi
77@105@99@114@111@115@111@102@116@         = Microsoft
119@110@104@101@108@112@46@100@108@108@    = wnhelp.dll
```

Table 10 - Micro version configuration values

C&C commands are sent via the PHP page. The Micro process parses these commands and runs the appropriate function. A file named 'prdata.sys' contains information about the infected host such as the MAC address, computer name and user name. Another file 'sdata.sys', located in the same folder, contains the last C&C server active path. Both of these files are stored encoded.

Micro also creates two other temporary files, 'systmp.dat' and 'systmp2.dat', in the %temp% folder.

Micro uses the same hardcoded User-Agent value as the other versions, and uses the same command line arguments '-i' and '-d' to install as service and kill the malware, respectively.

APPENDIX B – REMOVABLE MEDIA INFECTION

Explosive has integrated functionality to enable USB and other mass storage device infection. The functionality can be enabled or disabled by setting the DLD-USI flag in the configuration section.

When enabled, an additional configuration option, DLD-USA, dictates the specific infection method. The possible infection methods are:

- **Autorun.exe** – *Explosive* copies itself into the USB root directory and changes the filename to 'autorun.exe.'
- **Autorun.inf** – This is the same as the 'autorun.exe' option, but with an additional 'autorun.inf' file copied into the same directory.
- **EXE infection** – *Explosive* scans all *.exe files located in the USB drive, looking for previous infections. Previous infections are located by using the Exported PathProcess function from the *Explosive* DLL. If no previous infections were found, *Explosive* copies each *.exe file into the system's temporary folder (%temp%) and adds both the *Explosive* EXE and DLL files to its binary data To extract the injected files, a "Loader" binary is then injected into the file's binary. This "Loader" is set to be the main executable module. Once the injected file is executed, the "Loader" code is used to extract the *Explosive* files and resume the functionality of the original file. After the infection is complete, all infected files are returned to the USB drive and overwrite the original file. A special string "^!#^~|" is used by the "Loader" to parse and run the executable file.
- All – Uses all of the listed options.

APPENDIX C – INDICATORS OF COMPROMISE

Host Based IOCs

Service Names

Explosive can be installed with the following service names. The service is usually installed with no description value.

Possible *Explosive* Service Names
Helper
WindowsHelper
VMWareActivationHelper
WindowsInet
WindowsHelpService
WindowsHelpServices
WindowsInetService
MicrosoftIserv
MicrosoftServices
MicrosoftSystemClock

Main Module Filenames

These are the possible main *Explosive* modules filenames:

Possible Main Module Filenames	
aqagent.exe	vsmss.exe
qsagent.exe	w3wp.exe
cvsc.exe	whelp.exe
dllhost.exe	whttpd.exe
dllvhost.exe	winet.exe
dwcm.exe	winhelp.exe
embedded.exe	winhlp.exe
ieservice.exe	winhttpd.exe
logsys.exe	wininet.exe
nsp.exe	winlog.exe
rundll32.exe	winscr.exe
sccsc.exe	winscrv.exe
svchost.exe	winserv.exe
svsc.exe	wisrv.exe
svskey.exe	wnhelp.exe
syslog.exe	wnsys.exe
syswin.exe	wshelp.exe
updater.exe	wvsys.exe
vmacthlpsrv.exe	whelp.exe
vmtools.exe	whttpd.exe
vmtoolsd.exe	

DLL Filenames

These are the possible *Explosive* DLL filenames and the versions in which they appear:

Possible DLL filenames	*Explosive* Version
vsystem.dll	Version 3
winsec.dll	Version 2
tools.dll	Version 1
serverhelp.dll	KS version
wnhelp.dll	Micro version

Installation Paths

Explosive variants are installed and run under the following paths:

Possible Working Paths
%systemroot%
%systemroot%\system32
%systemroot%\SysWOW64
%appdata%
%programfiles%\VMware\VMware Tools
%programfiles%\VMWare\VMware Tools\win32
%programfiles%\Notepad++

Additional Paths

During its operation, *Explosive* uses several other files and directories for various tasks such as storing keylog data and other information extracted from the victim's system. The existence of these files and paths in a system can be used as an indicator of compromise.

These files and paths are most commonly set with "system" and "hidden" attributes.

Filename\Path
%systemroot%\Microsoft Help\Secure
%systemroot%\Microsoft Help\Secure\[username].tp.dat
%systemroot%\Microsoft Help\Secure\[username].tc.dat
%systemroot%\Microsoft Help\Secure\wintp\
%systemroot%\Microsoft Help\Secure\wintc\
%systemroot%\Microsoft Help\Secure\wintp\[username]-[date.time].dat
%systemroot%\Microsoft Help\Secure\wintc\[username]-[date.time].dat
c:\recycler\Microsoft Help\Secure
c:\recycler\Microsoft Help\Secure\[username].tp.dat
c:\recycler\Microsoft Help\Secure\[username].tc.dat
c:\recycler\Microsoft Help\Secure\wintp\
c:\recycler\Microsoft Help\Secure\wintc\
c:\recycler\Microsoft Help\Secure\wintp\[username]-[date.time].dat
c:\recycler\Microsoft Help\Secure\wintc\[username]-[date.time].dat
[CurrentRunningFolder]\[username]-rpt.sys
[CurrentRunningFolder]\[username]-crpt.sys
[CurrentRunningFolder]\winrpt
[CurrentRunningFolder]\wincrpt
[CurrentRunningFolder]\winrpt\[username]-[date.time].sys
[CurrentRunningFolder]\wincrpt\[username]-[date.time].sys

Network Based IOCs

C&C Updater Paths

Several URIs are used by both the dynamic and static C&C update servers. These are the observed values:

Possible C&C Updater URIs	
/ex/ie.php	/v2/p5/80/index.php
/445/ie.php	/v2/p5/443/index.php
/microsoft/ie.php	/v2/p5/445/index.php
/microsoft/index.php	/v2/p3/80/index.php
/80/index.php	/v2/p3/443/index.php
/443/index.php	/v2/p3/445/index.php
/445/index.php	/v3/80/index.php
/v2/443/index.php	/v3/443/index.php
/v2/445.index.php	/v3/445/index.php

C&C TCP Values

The detection of the following strings at the beginning of the TCP payload indicates a connection with the *Explosive* C&C server:

TCP Payload Starts With	Version
==gKg5XI+BmK	Version 2 and 3 (communication to and from the C&C server)
<*`!Q@W#E4'*>	Version 1 (communication to the C&C server)
<'|'>Explosive	Version 1 (communication to the C&C server)

HTTP Values

The C&C static and dynamic updaters both use HTTP for communication. While some of the following indicators are more common than others, they can all be used to detect *Explosive* C&C update communication:

HTTP Field	Value
User Agent	Mozilla/4.0 (compatible; MSIE 7.0; MSIE 6.0; Windows NT 5.1; .NET CLR 2.0.50727)
URL Contains	php?win=1
URL Contains	php?win=4
URL Contains	Php?micro=

Server Infrastructure

C&C Servers

These C&C server addresses are hardcoded in the various *Explosive* binaries:

IP Address	Geographical Location
69.64.90.94	*USA*
50.60.129.74	*USA*
85.25.20.27	*Germany*
213.204.122.130	*Lebanon*
213.204.122.133	*Lebanon*
184.107.97.188	*Canada*
69.94.157.80	*USA*

Static and Dynamic C&C Updater Servers

These domain names are used by the static C&C updater servers:

IP Address	Registered Info
saveweb.wink.ws	GoDaddy
carima2012.site90.com	GoDaddy
explorerdotnt.info	N/A
dotnetexplorer.info	Cloud Group Limited
dotntexplorere.info	Fastdomain inc.
xploreredotnet.info	N/A
erdotntexplore.info	Fastdomain inc.

SSH Server List

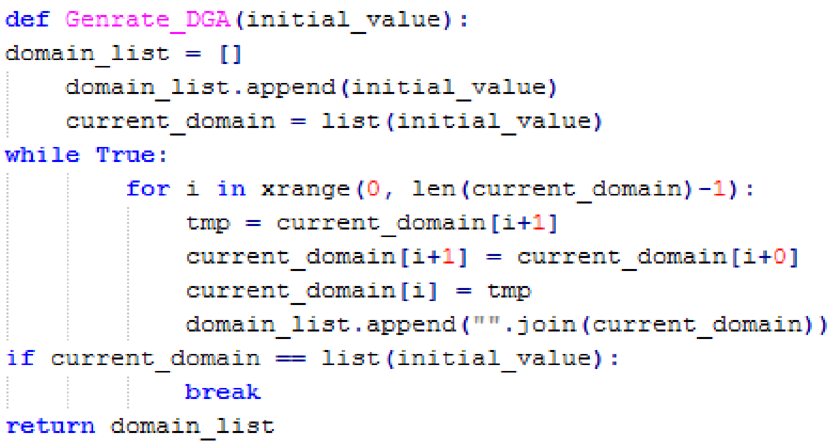
These IP addresses were detected as PLink servers used by the attacker for the SSH tunnel destinations:

IP Address	Geo Location
69.94.157.80	USA
50.60.129.78	USA

APPENDIX D – SCRIPTS AND SIGNATURES

Dynamic C&C Updater DGA Algorithm

The following Python script can be used to generate the results of the dynamic C&C updater DGA algorithm:



```
def Genrate_DGA(initial_value):
domain_list = []
    domain_list.append(initial_value)
    current_domain = list(initial_value)
while True:
        for i in xrange(0, len(current_domain)-1):
            tmp = current_domain[i+1]
            current_domain[i+1] = current_domain[i+0]
            current_domain[i] = tmp
            domain_list.append("".join(current_domain))
if current_domain == list(initial_value):
            break
return domain_list
```

These YARA signatures can be used to detect all versions of *Explosive* EXE and DLL files:

```
rule explosive_exe
{
  meta:
    author = “Check Point Software Technologies Inc.”
    info = “Explosive EXE”

  strings:
    $MZ = “MZ”
    $DLD_S = “DLD-S:”
    $DLD_E = “DLD-E:”

  condition:
    $MZ at 0 and all of them
}
```

```
import “pe”
rule explosive_dll
{
 meta:
   author = “Check Point Software Technologies Inc.”
   info = “Explosive DLL”

 condition:
   pe.DLL
   and ( pe.exports(“PathProcess”) or pe.exports(“_PathProcess@4”) ) and
pe.exports(“CON”)
}
```

APPENDIX E – OTHER INFORMATION

Complete List of C&C Commands:

The following is a complete list of the available C&C commands:

Encoded Command	Decoded Command	Description
'==gKg5XI+BmKqwUazRHUy92YlN3c'	ListProcess	List all running processes.
==gKg5XI+BmKqsUasxGUy92YlN3c	KillProcess	Kill a specified process.
==gKg5XI+BmKqIVduNUbkpCf	RunCmd	Run a specified command line.
==gKg5XI+BmK==gKF5WdttUZ5NnK	*EnumKeys*	Get the registry keys under a specified path.
==gKg5XI+BmK=oSRuVXbS92b0tUZ5NnK	*EnumRootKeys*	Get root registry keys.
==gKg5XI+BmK==gKHVGdSV2ZWFGb1VmK	GetRegValue*	Get a specified registry value.
==gKg5XI+BmKqQVZs5WZ0pCP	*`~!~`**Telnet*<	Connect remotely.
==gKg5XI+BmKqEEZkRUaypCP	*AddDir*<	Create a specified directory.
==gKg5XI+BmKqQUZsRUaypCP	*DelDir*	Delete a specified directory.
==gKg5XI+BmK==gKHVGdEJXa2V2cG9GbkVmc	*GetDrivesFolder	Get the content of a specific folder.
==gKg5XI+BmKqcUZ0RkcpZXZzpCP	*GetDrives*<	Get the drive list.
==gKg5XI+BmKqcUZ0ZUasVmK	`**GetFile*	Send a specific file to C&C server.
==gKg5XI+BmK=oyUjNFavRnK	*ScShot*	Get a screenshot.
==gKg5XI+BmK==gKEVXbwBVYzNnK	*DumpPass*	Dump saved passwords.
==gKg5XI+BmK==gKEVXbwhUazRnK	*DumpHist*	Dump IE history.
==gKg5XI+BmK=oySllHTvdmK	*KeyLog*	Get latest key logging file content.
==gKg5XI+BmK=oyQslGci9WYyRGTvdmK	*ClipboardLog*	Get latest clipboard logging file content.
==gKg5XI+BmK==gKF5WdtdVauR2b3NnK	*EnumWindows*	List open windows.
==gKg5XI+BmK=oCRlxWZ0VmRpxWZzpCP	*DeleteFiles*	Delete specified files.
==gKg5XI+BmK=oyQvBXeQF2c0VmRpxWZzpCP	*CopyPasteFiles*	Copy and paste specified files.
==gKg5XI+BmK==gKDVHdQF2c0VmRpxWZzpCP	*CutPasteFiles*	Cut and paste specified files.
==gKg5XI+BmKqoVawpCP	*Zip*<	Compress a specified file.
==gKg5XI+BmK=oSVupVawpCP	*UnZip*	Decompress a specified file to folder.
==gKg5XI+BmKqwUazRHUy92YlN3c	ListProcess	List all running processes.
==gKg5XI+BmKq8Ecl5GUGpyW	OpenPF	Open a specified file.
==gKg5XI+BmK==gKqMEbvNXZGlGblpiK	*CloseFile*	Close a specified file.
==gKg5XI+BmK=oiRpxWZTVmbkpCP	*FileSend*<	Send a specified file.
'==gKg5XI+BmK=wTIqIVRSVlTqEiP	<!*RERUN*!>	Restart *Explosive* process.

==gKg5XI+BmK==APhoySJxETqEiP	<!*KILL*!>	Kill *Explosive* process.
==gKg5XI+BmK8EiKEVETqEiP	*<!*DEL*!>	Kill *Explosive* process and remove all traces.
==gKg5XI+BmK8oCYF9kRgpiP	<*`EOF`*>	End of transmitted file.
==gKg5XI+BmK=wTlqaqEiP	<!*ok*!>	Confirm receipt of data.

Table 11 - Complete list of C&C commands

APPENDIX F – WEB SHELLS

The web shells injected into the compromised web servers are mostly custom made. They are written in various languages, such as ASP, ASP.Net and PHP.

These web shells contain many capabilities and have been seen to be heavily used by the attacker throughout the attack lifetime. Some of the web-shells functionalities are:

- Run remote commands
- Upload\Download files
- Account brute forcing
- Registry Access

The most common web shell used by the attackers is the *Caterpillar* web shell (name taken from the web shell code) which is a variant of the AspxSpy web shell.

Other web shells have also been used in the *Volatile Cedar* campaign, such as the KIDO web shell.

These are the filenames and hashes of the detected web shells:

File Name	MD5 Hash
404.asp	44db62acf787be73dcf8968d360f32b8
404.aspx	9f98eb473d3723f09d6a94cb326d4984
caterpillar.aspx	dab2cbb34ec587587bdf0418f7fb06b1
Heblib140201.aspx	d028eacd721e0b2d6e9ce19d2575d51b

APPENDIX G – SAMPLE HASHES

These sample hashes were seen during our analysis of the campaign:

MD5 Hash
eb7042ad32f41c0e577b5b504c7558ea
44b5a3af895f31e22f6bc4eb66bd3eb7
08c988d6cebdd55f3b123f2d9d5507a6
61b11b9e6baae4f764722a808119ed0c
c7ac6193245b76cc8cebc2835ee13532
184320a057e455555e3be22e67663722
5d437eb2a22ec8f37139788f2087d45d
1dcac3178a1b85d5179ce75eace04d10
9a5a99def615966ea05e3067057d6b37
2b9106e8df3aa98c3654a4e0733d83e7
ab3d0c748ced69557f78b7071879e50a
c9a4317f1002fefcc7a250c3d76d4b01
4f8b989bc424a39649805b5b93318295
3f35c97e9e87472030b84ae1bc932ffc
7cd87c4976f1b34a0b060a23faddbd19

ea53e618432ca0c823fafc06dc60b726
034e4c62965f8d5dd5d5a2ce34a53ba9
5ca3ac2949022e5c77335f7e228db1d8
306d243745ba53d09353b3b722d471b8
e6f874b7629b11a2f5ed3cc2c123f8b6
5b505d0286378efcca4df38ed4a26c90
7dbc46559efafe8ec8446b836129598c
1d4b0fc476b7d20f1ef590bcaa78dc5d
66e2adf710261e925db588b5fac98ad8
c898aed0ab4173cc3ac7d4849d06e7fa
22872f40f5aad3354bbf641fe90f2fd6
c19e91a91a2fa55e869c42a70da9a506
740c47c663f5205365ae9fb08adfb127
edaca6fb1896a120237b2ce13f6bc3e6
d2074d6273f41c34e8ba370aa9af46ad
6f11a67803e1299a22c77c8e24072b82
7031426fb851e93965a72902842b7c2c
981234d969a4c5e6edea50df009efedd
2783cee3aac144175fef308fc768ea63
f58f03121eed899290ed70f4d19af307
96b1221ba725f1aaeaaa63f63cf04092
29eca6286a01c0b684f7d5f0bfe0c0e6
826b772c81f41505f96fc18e666b1acd

APPENDIX H – CHECK POINT DETECTION NAMES

Name
Trojan.Win32.Explosive
Trojan.Win32.Explosive.A
Trojan.Win32.Explosive.B
Trojan.Win32.Explosive.C

ADDITIONAL INFORMATION

The information in this report is based on partial visibility and evidence collected during our investigation.

The Volatile Cedar investigation is still ongoing. We hope to release further information in upcoming reports.

If you suspect you were targeted by this campaign, or can share additional information on this campaign based on other meaningful observations please contact volatilecedar@checkpoint.com