context
INFORMATION SECURITY

Context Threat Intelligence

Threat Advisory

The Monju Incident

Context Ref.	TA10009
Author	Context Threat Intelligence (CTI)
Date	27/01/2014
Tel	+44 (0) 20 7537 7515
Fax	+44 (0) 20 7537 1071
Email	threat@contextis.co.uk

Contents

1 Distribution

Context Information Security distribute Context Threat Intelligence (CTI) reporting under the Traffic Light Protocol (TLP)[1], a method of classifying a document in order to promote the distribution of sensitive information between individuals, organisations or communities in a controlled and trusted way, based on the originator's wishes.

The various levels of the TLP are represented by the following colours:

RED - Personal; for named recipients only

Sources may use TLP: RED when information cannot be effectively acted upon by additional parties, and could lead to impacts on a party's privacy, reputation, or operations if misused.

Recipients may not share TLP: RED information with any parties outside of the specific exchange, meeting, or conversation in which it is originally disclosed. TLP: RED information will be passed verbally or in person.

AMBER - Limited Distribution

Sources may use TLP: AMBER when information requires support to be effectively acted upon, but carries risks to privacy, reputation, or operations if shared outside of the organisations involved.

Recipients may only share TLP: AMBER information with members of their own organisation who need to know, and only as widely as necessary to act on that information.

GREEN – Community Wide

Sources may use TLP: GREEN when information is useful for the awareness of all participating organisations as well as with peers within the broader community or sector.

Recipients may share TLP: GREEN information with peers and partner organisations within their sector or community, but not via publicly accessible channels such as publication or posting publicly on the Internet.

WHITE - Unlimited Distribution

Sources may use TLP: WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release.

Subject to standard copyright rules, TLP: WHITE information may be distributed freely, without restriction.

[1] http://www.oecd.org/dataoecd/25/10/40761118.pdf

2 Executive Summary

On 2nd January 2014 a Systems Administrator at the Monju fast breeder reactor facility in Japan noticed suspicious connections emanating from a machine in the control room, coinciding with what was a seemingly routine software update to a free media player. Among other items, staff training documents and more than 40,000 emails were stored on the machine and thought to be harvested by the attacker. The Japanese Atomic Energy Agency is investigating further.

The attack appears to have been the result of the attackers having compromised the 'GOM Player' update server and having it act as a 'watering hole', meaning that machines which access the site are delivered malware. Gom Player originates in South Korea and in some parts of Asia it is a popular alternative to Windows Media Player. It is unclear whether every machine trying to download an update received this malware or whether only machines which fitted a certain profile were infected.

Technical analysis of the implant on the compromised machine has shown it to be a variant of a Trojan which has been in the wild for some years now and continues to be effective. The 'Gh0st RAT' has been used extensively in attacks linked to the Chinese state, though it is important to remember that the code is publicly available and can be modified and used by anyone. The targeting of a Japanese nuclear facility however, is consistent with Chinese state intelligence requirements. If this is the work of a Chinese group then we feel the targeting may go much further than the Civil Nuclear sector and thus be of interest to the wider Energy Sector and industry as a whole.

In order to inform the Energy Sector and beyond about this attack, we have compiled a technical summary of the attack and have provided a number of Indicators Of Compromise (IOCs) which can be used to aid detection. It is likely that the attackers would redeploy their implant against other targets, albeit with a delivery mechanism more tailored to the location of the intended victims.

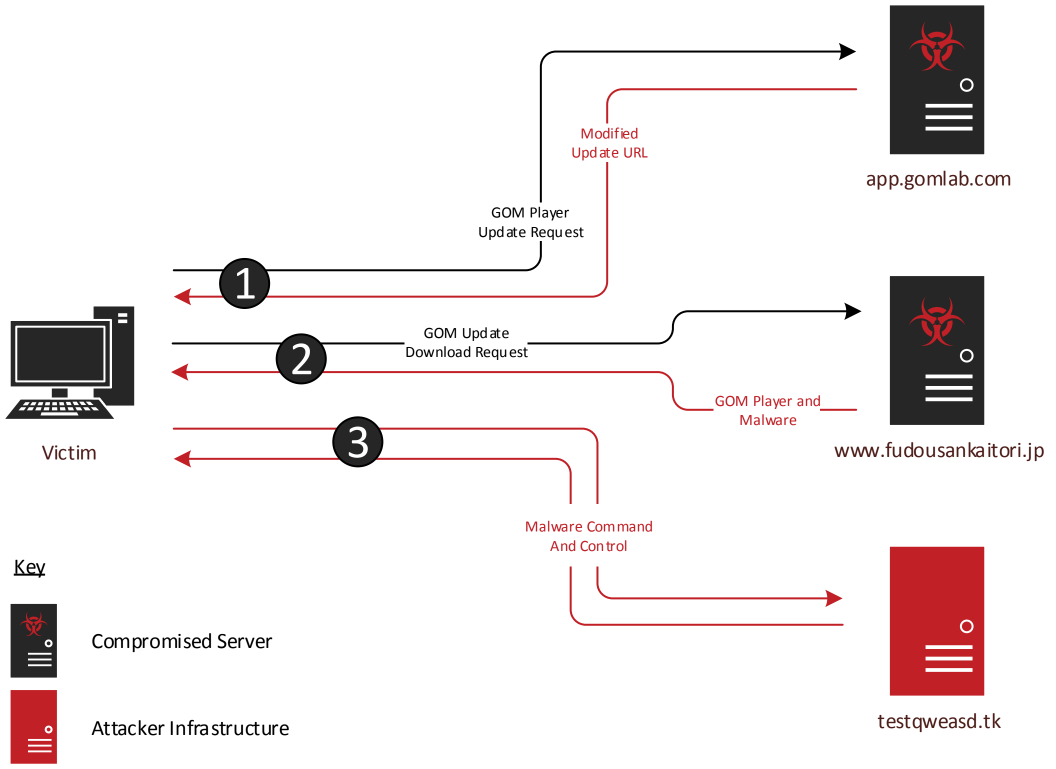
3 The Monju Incident

3.1 Infection Vector

Based on open source reporting, it appears that the intrusion took place via the compromise of the GOM Player update server (app.gomlab.com), where attackers may have gained entry via a PHP-based webshell, hidden within an image, present on the host since October 2011[2].

The observed malicious activity relates to the modification of a file that controls GOM Player updates, spanning the date range 27th December 2013 to 16th January 2014, during which time these alterations are reported to have only manifested themselves for visitors on certain IP ranges; evidence supporting this claim has not yet been made public. If this was indeed the case, then the nature of this attack is certainly more targeted than one that would cover the entire userbase of the GOM Player product, with victims comprising of the Japanese Government in addition to those at the Monju nuclear facility.

The modified file redirected the GOM Player update process to another compromised server (www.fudousankaitori.jp (203.189.101.35)), where a file containing both the legitimate update and the malware was deliver to the victim.



A diagram illustrating the modified flow of the GOM Player update process which led to the compromise

[2] http://hummingbird.tistory.com/5187

3.2 Malware

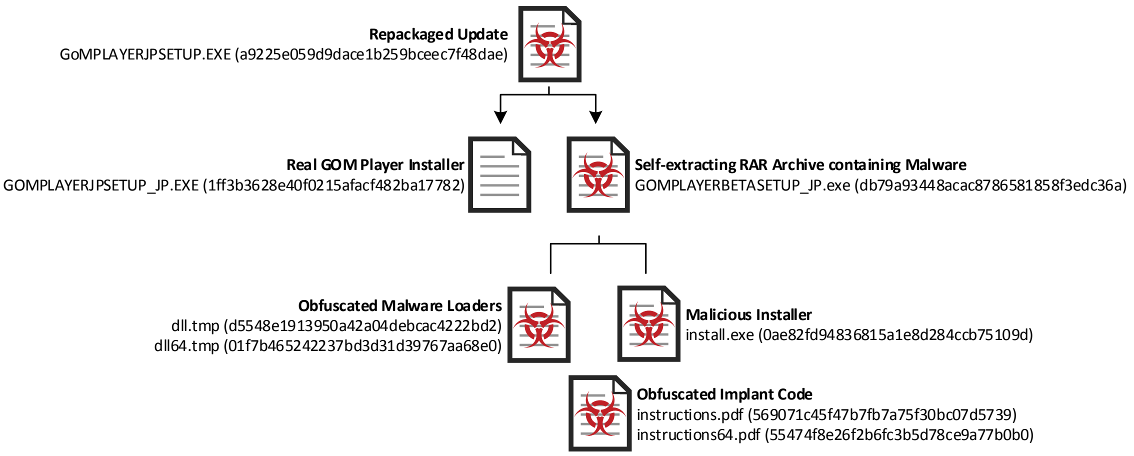
3.2.1 Overview

Deployed to the system via a compressed bundle containing the official GOM Player setup binary and a self-extracting RAR archive containing the malicious files, the malware consists of a number of individual pieces. Upon extraction from the RAR archive, the installer component (`0ae82fd94836815a1e8d284ccb75109d`) is automatically launched alongside the GOM Player update, distracting victims from the malicious activity taking place.

The installer component is referred to by the author as 'miansha' which, according to an East Asia Cyber Threat Intelligence Researcher, is likely Miǎnshā (免杀), a phrase commonly used by People's Republic of China (PRC) hackers to mean 'anti Antivirus detection' or 'Antivirus avoidance'; Symantec[3] have named the detection for this code 'Backdoor.Miancha', where Miǎnchá (免查, likely shorthand for 免杀查) similarly means 'Antivirus avoidance'. The installer is responsible for the malware persistence mechanism, adding entries to the registry in the following locations, depending on Windows Version:

Miancha Persistence Registry Keys	
Windows Vista and later	`HKEY_USERS\.default\Software\Classes\CLSID\{ECD4FC4D-5213-11D0-B792-00A0C90312E1}\InProcServer32\@ = expand:"C:\WINDOWS\temp\install.ocx"`
Prior to Windows Vista	`HKEY_USERS\.default\Software\Classes\CLSID\{B12AE898-D056-4378-A844-6D393FE37956}\InProcServer32\@ = expand:"C:\WINDOWS\temp\install.ocx"`

The installer will also determine the system architecture (32- or 64-bit) and then deobfuscate the relevant loader DLL to the path 'C:\Windows\temp\install.ocx', ensuring the malware is launched on system start-up. Oddly, this file is padded with null bytes, resulting in a 25 megabyte file.


Repackaged Update
GoMPLAYERJPSETUP.EXE (a9225e059d9dace1b259bceec7f48dae)
Real GOM Player Installer
GOMPLAYERJPSETUP_JP.EXE (1ff3b3628e40f0215afacf482ba17782)
Self-extracting RAR Archive containing Malware
GOMPLAYERBETASETUP_JP.exe (db79a93448acac8786581858f3edc36a)
Obfuscated Malware Loaders
dll.tmp (d5548e1913950a42a04debcac4222bd2)
dll64.tmp (01f7b465242237bd3d31d39767aa68e0)
Malicious Installer
install.exe (0ae82fd94836815a1e8d284ccb75109d)
Obfuscated Implant Code
instructions.pdf (569071c45f47b7fb7a75f30bc07d5739)
instructions64.pdf (55474f8e26f2b6fc3b5d78ce9a77b0b0)

The deployment chain of the Miancha Gh0st variant

[3] http://www.symantec.com/security_response/writeup.jsp?docid=2014-012407-3922-99

The main implant code is stored in files named instructions.pdf and instructions64.pdf; not PDF documents but instead DLLs obfuscated with a one-byte XOR with 0x14, similar to the malware loader DLLs. The loader, referred to by the malware author as 壳 (shell), reads and deobfuscates the main implant code which then communicates with the attacker-controlled server at testqweasd.tk (211.43.220.89) on TCP port 443. The main implant code is referred to as 白加黑 (‘Black on White’), a term used in the PRC hacking community to denote the act of Antivirus avoidance through the loading of malicious ‘black’ code via non-malicious or trusted ‘white‘ code. This is a practice recently illustrated through the deployment of the PlugX trojan, utilising DLL load order hijacking alongside a signed (trusted) executable.

Analysis of this malware revealed it to be a variant of the Gh0st RAT, often used by Chinese actors (including those who are state-motivated or directly state-sponsored). This specific variant shows similarities to that used during the VOHO campaign[4], where Gh0st RAT was spread via watering hole attacks utilising vulnerable websites belonging to financial services and technology companies. Specifically, the initial five bytes of the communications (often used to denote a campaign or victim) are ‘HTTPS’, amended from the original ‘Gh0st’; the same as the traffic produced by the VOHO Gh0st variant.

In addition to delivering system-specific details back to the attacker, Gh0st RAT provides the capability to deploy additional malware, enabling the harvesting of sensitive data and enabling the further propagation throughout the infected network.

[4] https://blogs.rsa.com/voho-apt-campaign-update/

3.2.2 Detection

To enable rapid response, the following Snort signature can be deployed:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 53,80,443,1080 (msg:"gh0st RAT 'HTTPS' variant
(aka Backdoor.Miancha)"; flow:established,to_server; content:"HTTPS"; depth:5; rawbytes;
classtype:trojan-activity; sid:xxx; rev:1;)
```

Additionally, the following Yara signature should identify both encoded payloads and the active implant in memory:

```
rule Trojan_W32_Gh0stMiancha_1_0_0
{
   strings:
       $0x = { 57 5b 5a 5a 51 57 40 34 31 67 2e 31 70 34 5c 40 40 44 3b 25 3a 19 1e 5c 7b
67 60 2e 34 31 67 2e 31 70 19 1e 55 77 77 71 64 60 2e 34 3e 3b 3e 19 1e 57 7b 7a 60 71 7a
60 39 40 6d 64 71 2e 34 60 71 6c 60 3b 7c 60 79 78 19 1e 44 66 7b 6c 6d 39 57 7b 7a 7a 71
77 60 7d 7b 7a 2e 34 5f 71 71 64 39 55 78 7d 62 71 19 1e 57 7b 7a 60 71 7a 60 39 78 71 7a
73 60 7c 2e 34 24 19 1e 19 1e }
       $1 = { 5c e7 99 bd e5 8a a0 e9 bb 91 5c }
       $1x = { 48 f3 8d a9 f1 9e b4 fd af 85 48 }
       $2 = "DllCanLoadNow"
       $2x = { 50 78 78 57 75 7a 58 7b 75 70 5a 7b 63 }
       $3x = { 5a 61 79 76 71 66 34 7b 72 34 67 61 76 7f 71 6d 67 2e 34 31 70 }
       $4 = "JXNcc2hlbGxcb3Blblxjb21tYW5k"
       $4x = { 5e 4c 5a 77 77 26 7c 78 76 53 6c 77 76 27 56 78 76 78 6c 7e 76 26 25 60 4d
43 21 7f }
       $5 = "SEFSRFdBUkVcREVTQ1JJUFRJT05cU3lzdGVtXENlbnRyYWxQcm9jZXNzb3JcMA=="
       $5x = { 47 51 52 47 46 52 70 56 41 7f 42 77 46 51 42 40 45 25 5e 5e 41 52 46 5e 40
24 21 77 41 27 78 6e 70 53 42 60 4c 51 5a 78 76 7a 46 6d 4d 43 6c 45 77 79 2d 7e 4e 4c 5a
6e 76 27 5e 77 59 55 29 29 }
       $6 = "C:\\Users\\why\\"
       $6x = { 57 2e 48 41 67 71 66 67 48 63 7c 6d 48 }
       $7 = "g:\\ykcx\\"
       $7x = { 73 2E 48 6D 7F 77 6C 48 }
       $8 = "(miansha)"
       $8x = { 3C 79 7D 75 7A 67 7C 75 3D }
       $9 = "server(\xE5\xA3\xB3)"
       $9x = { 7C 2E 48 26 24 25 27 3A 25 25 3A 26 21 48 67 71 66 62 71 66 3C F1 B7 A7 3D
48 46 71 78 71 75 67 71 48 67 71 66 62 71 66 3A 64 70 76 }
       $cfgDecode = { 8a ?? ?? 80 c2 7a 80 f2 19 88 ?? ?? 41 3b ce 7c ??}
   condition:
       any of them
}
```

4 Appendix A – File Metadata

Gh0stMiancha Installer	
MD5	0ae82fd94836815a1e8d284ccb75109d
SHA1	bcba2a4d55d860f0bca3b9f80a5deb2dd69f000c
SHA256	b2f9e2f7c07235a6ea03e90ba591f0a43d38d8ff8ee6583473b6fbb63831619d
Size (bytes)	13314
Compile Time	2013-11-22 12:19:48 UTC
In-the-wild Filenames	install.exe
PDB String	g:\ykcx\install(miansha)\Release\install.pdb

Obfuscated TrojanLoader:W32/Gh0stMiancha	
MD5	d5548e1913950a42a04debcac4222bd2
SHA1	ac48bc2deefd30dad762a23e85409a7eec48b723
SHA256	3d43f7fab3c8f574e2790c2d97f85fa87f0d53e412c995462e53348b4fc34b74
Size (bytes)	10299
Compile Time	N/A
In-the-wild Filenames	dll.tmp

TrojanLoader:W32/Gh0stMiancha	
MD5	04e7361323b431f7c9f86388f316bbea
SHA1	e3c095c7ace563b41b3f4310f3de69e47c86fd03
SHA256	73ef70f1e80e32341eebcb3b1084cf896f6b1aa701b7a6c7abcb9293500d84ae
Size (bytes)	10299
Compile Time	2013-11-26 09:34:10 UTC
In-the-wild Filenames	install.ocx
PDB String	h:\2013.11.25\server(壳)\Release\server.pdb

Obfuscated TrojanLoader:W64/Gh0stMiancha	
MD5	01f7b465242237bd3d31d39767aa68e0
SHA1	db4ec59bf7f34a21f9dc7f2ded68c616f7c0fe47
SHA256	ed39c1d86ff8cfe18ef58e850d205a678d255150324b00661b91448173c94900
Size (bytes)	12347
Compile Time	N/A
In-the-wild Filenames	dll64.tmp

TrojanLoader:W64/Gh0stMiancha	
MD5	008fbd0fde06edb31fc7eecdae1a3030
SHA1	b9ae0a079cd1dae96425ced4bb96ba0f71c87a7a

SHA256	cc8d38d3cc214ff3ad10d6859a88e018b1f7e0ed6df7d04a6f4368bda851ba14
Size (bytes)	12347
Compile Time	2013-11-26 11:47:39 UTC
In-the-wild Filenames	install.ocx
PDB String	C:\Users\why\Desktop\server(壳)\x64\Release\server.pdb

Obfuscated Trojan:W32/Gh0stMiancha	
MD5	569071c45f47b7fb7a75f30bc07d5739
SHA1	540bb9d2dee8f4e10e5ae0a5cc900b346a57a198
SHA256	8a00b2aefdcd0bb22013bbe9c7941fa16af8246e545e1522622006b9c88ca716
Size (bytes)	169019
Compile Time	N/A
In-the-wild Filenames	instructions.pdf

Trojan:W32/Gh0stMiancha	
MD5	916b1a07efb145c450b4c13540be6c3e
SHA1	7984639beb4e9870301d3b44a68b4346f9a6b826
SHA256	f26c2e9bee680f8e4d7afd73e2984a6697263334d2f0049a40e050d75293035e
Size (bytes)	169019
Compile Time	2013-12-06 08:08:28 UTC
In-the-wild Filenames	N/A
PDB String	h:\2013.11.25\白加黑\server(update.dll)(instructions.pdf)\Release\server.pdb

Obfuscated Trojan:W64/Gh0stMiancha	
MD5	55474f8e26f2b6fc3b5d78ce9a77b0b0
SHA1	3f714c33992e906e69df2d5d4971beaed336d9f4
SHA256	27e5670f68ff68acc80716c6870f4e5d06c471791f087d5b9b7613f8dc700037
Size (bytes)	233019
Compile Time	N/A
In-the-wild Filenames	instructions64.pdf

Trojan:W64/Gh0stMiancha	
MD5	1d2c77f0f8a715de09ce6fae5fc800d4
SHA1	30784735763b060a39f76c29439a6aebbf6a4b9b
SHA256	2fdf454f6b1c82d757d054bea5f0438f5da1ecd9e5059610d3d4b74e75a7c8b0
Size (bytes)	233019
Compile Time	2013-12-06 08:10:34 UTC
In-the-wild Filenames	N/A
PDB String	C:\Users\why\Desktop\server(update.dll)(instructions.pdf)x64\x64\Release\server.pdb

Context Information Security - Threat Intelligence - threat@contextis.co.uk

London (HQ)

4th Floor
30 Marsh Wall
London E14 9TP
United Kingdom

Cheltenham

Corinth House
117 Bath Road
Cheltenham GL53 7LS
United Kingdom

Düsseldorf

1.OG
Adersstr. 28
40215 Düsseldorf
Germany

Melbourne

4th Floor
155 Queen Street
Melbourne VIC 3000
Australia