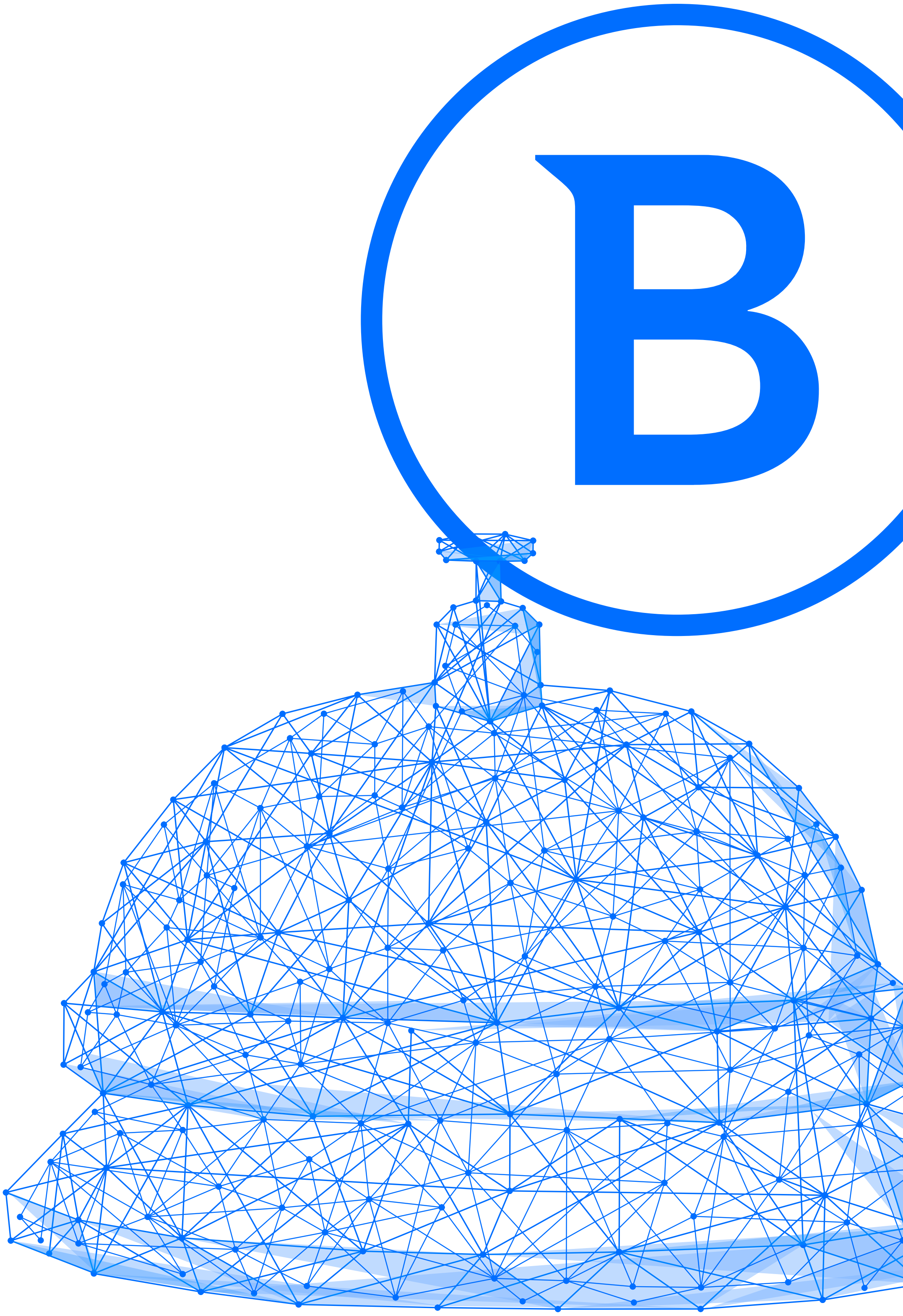
Bitdefender

WHITEPAPER

Booking Engine Security Revisited - the Anatomy of a Cyber-Attack against Hotels



Contents

Authors:

Alexandru MAXIMCIUC – Principal Forensics Engineer, Cyber Threat Intelligence Lab @ Bitdefender

Adrian SCHIPOR - Security Researcher, Cyber Threat Intelligence Lab @ Bitdefender

Victor VRABIE - Security Researcher, Cyber Threat Intelligence Lab @ Bitdefender

Foreword

Booking engines – they make the worlds of travel and hospitality spin around. Estimated at over $US 500 billion, this market moves fast. These engines are a critical, nearly invisible part of the hospitality industry, and their security is essential to protect guests' personal and financial information. Occasionally, booking technology falls victim to motivated threat actors who use vulnerabilities in code to get access to sensitive customer information such as name, address, email address, phone number, credit or debit card number, expiration date, and security code or card verification code.

This was the case of a cyber-attack discovered back in 2021 against the IRM Next Generation online booking engine built by Resort Data Processing, Inc. ("RDP"). This attack is probably not singular amongst the wide range of online booking engines built by various other software companies. However, it is closely related to an investigation that Bitdefender was called in for help. Incidentally, the results of the investigation also helped us understand how the 2021 cyber-attack against IRMNg took place and we're drafting our findings in this report to help other business entities stay protected.

Note: As a global cyber-security player, we understand the importance of responsible disclosure. In this spirit, we have spent more than 90 days attempting to get in touch with the vulnerable vendor. Our attempts to establish first contact have gone unanswered, while cyber-criminals continue to use these tactics against unaware victims. After careful consideration, we have decided to publicly release limited information about these vulnerabilities and let IRMNg users know about them.

In November 2022, Bitdefender researchers in the Cyber-Threat Intelligence Lab have started investigating signs of suspicious activity on a server owned by a resort in the United States of America, when files part of the booking engine developed by Resort Data Processing were illegally accessed by a third-party. Our initial assessment revealed the presence on the server of several webshell components, as well as of a variant of MicroBackdoor. We were able to also isolate a malicious IIS native module with backdoor functionalities called XModule, which was specially designed for e-skimming (theft of credit card information and passwords by injecting malicious code in a JavaScript file used by Resort Data Processing's IRMNg booking engine).

This stealer component of XModule is specifically developed for instances of IRMNg booking engine, as it is injecting the malicious JavaScript in a file used by this booking engine. Our assessment is based on the fact that its name was hardcoded in the samples we collected during the forensic exercise. This component can also act as a proxy between MicroBackdoor and the C2.

Our analysis indicates that the infection started in the summer of 2022, but it seems that the attackers used timestomping (a technique that modifies the timestamps of a file to blend it in with other legitimate files in the same folder) to make some samples look like they were created a few years before. While we can't confidently identify the threat actor group behind the attack, we are certain that the purpose of the attack is financial gain and theft of personal information. We found several other victims using the same booking engine that were infected with similar webshells, but we couldn't isolate the XModule component on any other victim.

The infection vector could not be exactly determined, but there are artefacts that suggest that the initial compromise avenue on the server was the IRMNg booking engine:

- ↳ Several webshells were located on IRMng's default file upload directory
- ↳ a custom tool was executed to run PSQL queries on the database used by the booking engine
- ↳ some of the first commands executed in the attack were related to a service that is part of the IRMNg booking engine and were used for privilege escalation

As an observation, all these artefacts have shown that the threat actors were very familiar with the internals of the booking engine software.

We began to analyze the booking engine and found multiple vulnerabilities; notably an unauthenticated file upload vulnerability (CVE-2023-39424) that seems to have been fixed at some point. We also found vulnerabilities that are currently not fixed, including a flaw in the authentication process by using a special account that allows Resort Data Process employees to log on to their clients' management interfaces and APIs, this account having a **daily** password that can be easily generated by analyzing a specific DLL . We must note that our goal during this investigation was not to fully assess the security of the booking engine, but to establish if this could have been the entry point, so there may be vulnerabilities that we missed.

Attack at a glance

- ↳ While investigating anomalous activity, Bitdefender researchers found malicious files on servers running the IRM Next Generation online booking engine built by Resort Data Processing, Inc.
- ↳ Our investigation reveals the extent of the attack but also outlines several vulnerabilities in the IRM Next Generation online booking engine that were identified, catalogued and responsibly reported to the vulnerable vendor as per the timeline below.

Identified vulnerabilities

- **CVE-2023-39420** - Use of Hard-coded Credentials in RDPCore.dll (CWE-798)
- **CVE-2023-39421** - Use of Hard-coded Credentials in RDPWin.dll CWE-798)
- **CVE-2023-39422** - Use of Hard-coded Credentials in /irmdata/api/ endpoints (CWE-798)
- **CVE-2023-39423** - Improper Neutralization of Special Elements used in an SQL Command in RDPData.dll (CWE-89)
- **CVE-2023-39424** - Improper Neutralization of Special Elements in Output Used by a Downstream Component (‘Injection’) in RDPngFileUpload.dll (CWE-74)

Disclosure timeline

April-May, 2023 – Bitdefender identifies issues in multiple components of the IRMNg application during a malware infection investigation

May 23, 2023 – Bitdefender makes a first contact attempt with the vulnerable vendor via email

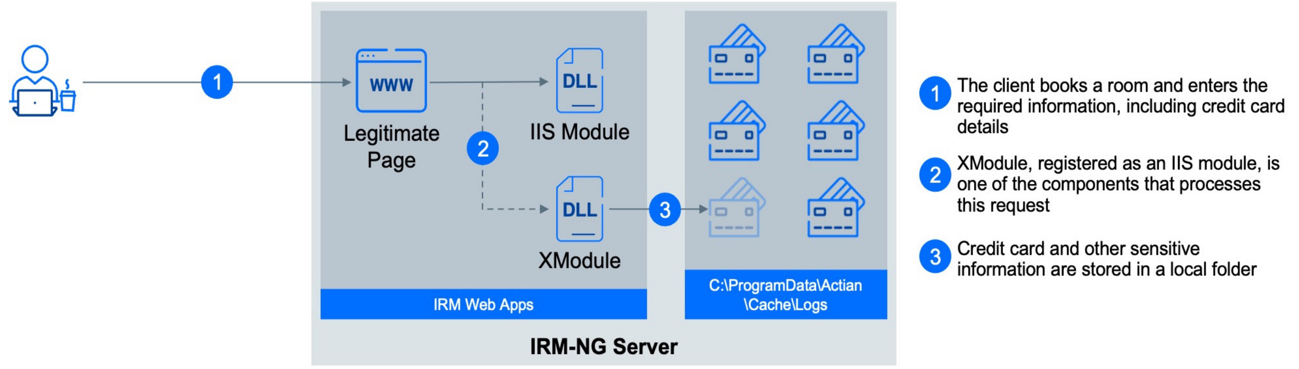
May 30, 2023 – Given that the previous attempt did not yield any result, Bitdefender makes a second attempt via email

August 02, 2023 – Bitdefender allocates CVE numbers for the identified vulnerabilities

August 16, 2023 – Bitdefender continues to reach out to the vulnerable vendor through Twitter, Facebook. Our efforts go once again unacknowledged

September 07, 2023 – This report becomes public as part of our responsible disclosure program

A technical analysis of the attack


1
WWW
Legitimate Page
2
DLL
IIS Module
DLL
XModule
3
IRM Web Apps
C:\ProgramData\Actian \Cache\Logs
IRM-NG Server
1 The client books a room and enters the required information, including credit card details
2 XModule, registered as an IIS module, is one of the components that processes this request
3 Credit card and other sensitive information are stored in a local folder

Initial compromise

The first command executed by the attackers was:

```
2022-07-12 07:36:17   cmd.exe /c systeminfo
```

Although we could not confirm it, we are confident that the initial infection vector was related to some zero-day vulnerabilities in Resort Data Processing’s booking engine IRMNg. This assumption is backed up by some of the first commands executed by the attackers:

```
2022-07-12 07:40:23   cmd.exe /c rename c:\\inetpub\\wwwroot\\rdprepository\\irm\\
                      content\\<SERVER>\\<RESORT>\\index.css index.aspx
```

```
2022-07-12 07:40:35  cmd.exe /c rename c:\\inetpub\\wwwroot\\rdprepository\\irm\\
                     content\\<SERVER>\\<RESORT>\\index.css index2.aspx
```

As shown, the attackers uploaded some webshells with the .css, extension - this being one of the extensions allowed by the file uploading API, and then renamed them to .aspx. The location of choice also reinforces the fact that they abused a flaw in the booking engine, as this is the default uploading destination for a management user that is **logged on** to a specific server/resort. The fact that the attackers were logged on to a management interface is also reinforced by a command executed soon after they renamed the malicious webshells:

```
2022-07-12 07:54:48  c:\\windows\\system32\\cmd.exe /c cd “C:\\Users\\<USERNAME>\\XRM\\Files\\” &&
                     dir && ConsoleApplication5.exe <SERVER> <DATABASE> “DELETE FROM Logins WHERE
                     Browser=’Firefox94’”
```

The ConsoleApplication5.exe tool is custom developed by the attackers to execute PSQL queries on the database (Pervasive PSQL) used by the booking engine (the database wasn’t password protected). We should note that the execution of these commands was possible only because the attackers already could execute commands with “cmd.exe /c”, which we believe is due to another vulnerability in the product (**CVE-2023-39424**).

The fact that this custom tool (ConsoleApplication5.exe) was used soon after the initial compromise (the tool was executed 18 min after the first command) also suggests that the attackers were aware of the internals of the booking engine before the compromise. The PE header confirms this as the tool was compiled on 2022-07-11 12:49:31.

Privilege Escalation

The same tool was also used to abuse a service used by IRMNg named “RDPng File Upload”, which is processing the “FileUploads” table (that contain the content and the destination of the files that should be uploaded) and writes the content at the specified path. Because the service is running as SYSTEM, the attackers abused it to write a malicious dll to a protected path, as it can be seen in the command lines below:

```
2022-07-12 07:56:12  c:\\windows\\system32\\cmd.exe /c cd “C:\\Users\\<USERNAME>\\XRM\\
                     Files\\” && dir && ConsoleApplication5.exe <SERVER> < DATABASE >
                     “INSERT INTO FileUploads (FileName,File,Action,Location,DateEntered)
                     VALUES (‘wow64log.dll’,’41’,0,’c:\\inetpub\\wwwroot\\rdprepository\\
                     irm\\content\\<SERVER>\\<RESORT>\\’,now());UPDATE FileUploads SET
                     File=(SELECT CAST(Description as char(5120)) FROM UnitAdContent WHERE
                     PropertyName=’333333333333333333333333333333’) Where FileName=’wow64log.dll’”
2022-07-12 07:56:57  c:\\windows\\system32\\cmd.exe /c cd “C:\\Users\\<USERNAME>\\XRM\\
                     Files\\” && dir && ConsoleApplication5.exe <SERVER> <DATABASE>
                     “INSERT INTO FileUploads (FileName,File,Action,Location,DateEntered)
                     VALUES (‘wow64log.dll’,’41’,0,’c:\\inetpub\\wwwroot\\rdprepository\\
                     irm\\content\\<SERVER>\\<RESORT>\\..\\..\\..\\..\\..\\..\\..\\..\\
                     windows\\system32\\’,now());UPDATE FileUploads SET File=(SELECT
                     CAST(Description as char(5120)) FROM UnitAdContent WHERE
                     PropertyName=’333333333333333333333333333333’) Where FileName=’wow64log.dll’”
```

The content of the malicious DLL was not provided at the command line, but taken from a record from the table named UnitAdContent (the one having PropertyName=’333333333333333333333333333333’)

The attacker also used other tools for privilege escalation: the well-known PrintSpoofer (though the sample was packed with Themida) and a POC for CVE-2020-0787 to copy files to a protected location (the POC is a modified version of the one that can be found in this repository).

Persistence and defense evasion

For persistence, the attackers used a scheduled task named “ChkUpd” that ran as SYSTEM. The role of the task was to execute a malicious DLL: C:\Windows\system32\rundll32.exe batchd.dll,ResChkUpd. As the DLL was written to a default location for DLLs (C:\Windows\SysWOW64\batchd.dll) the task executed the DLL without specifying the full path, thus being less suspicious. The DLL was very small and only executed a .bat file (“c:\\irmsetup\\install.bat”) with the role of deploying the malicious components. We suspect there were various .bat installers at different stages, but we found only one, with the following content:

```
copy “C:\Users\All Users\XRM\Data\api.dll” C:\windows\system32\logapi64.dll

copy “C:\Users\All Users\XRM\Data\wow.dll” C:\windows\system32\wow64log.dll

copy “C:\Users\All Users\XRM\Data\x.config” C:\inetpub\wwwroot\web.config

copy “C:\Users\All Users\XRM\Data\i.dat” C:\inetpub\wwwroot\rdprepository\irm\
content\<SERVER>\<RESORT>\index.aspx
```

The files are described in the table below:

Source Path	Description
C:\Users\All Users\XRM\Data\api.dll	Micro Backdoor
C:\Users\All Users\XRM\Data\wow.dll	MicroBackdoor Installer
C:\Users\All Users\XRM\Data\x.config	we suspect is an altered web config that loads XModule
C:\Users\All Users\XRM\Data\i.dat	.ASPX webshell

For defense evasion, we saw that the attackers used a process ghosting tool named KingHamlet (that can be found here), e.g.:

```
2022-07-18 00:02:50   c:\\windows\\system32\\cmd.exe /c cd C:\\temp\\tmp && procghost.
                      exe C:\\temp\\tmp\\PrintSpoofer64.exe.khe netservice test.exe -cmdline
                      PrintSpoofer64.exe -c C:\\temp\\tmp\\irm.bat

2022-08-04 10:01:33   c:\\windows\\system32\\cmd.exe /c cd C:\\temp\\tmp && procghost.
                      exe C:\\temp\\tmp\\PrintSpoofer64.exe.khe netservice test.exe -cmdline
                      PrintSpoofer64.exe -c "C:\\windows\\system32\\rundll32.exe C:\\windows\\
                      system32\\wow64log.dll DllMain"
```

They also used timestomping for various files, including webshells, the .bat installer, the Micro Backdoor and XModule executables.

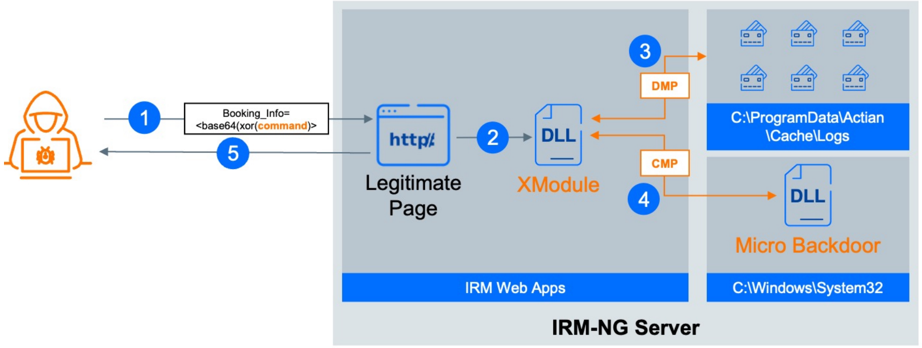
Execution

The goal of the infection was the theft of card information and passwords. To achieve this, the attackers deployed a malicious native IIS module named XModule and a sample of Micro Backdoor (with C2 pointed to telecomptd[.]org).

The interesting part is the fact that they could communicate through a named pipe, with XModule acting as a proxy between the C2 and Micro Backdoor, thus making both processes less suspicious, e.g.:

- the IIS module was working as designed: it was processing the request and serving the response body
- although the Micro Backdoor process was executing suspicious commands, it wasn't producing any network traffic

In order to send commands to the backdoor, the attacker made a POST request with a specific body to any (legitimate) page of the web server; the XModule would process the request and send the extracted command to the Micro Backdoor, getting the result and writing it back in the response body. This is an almost undetectable method of communication.



1 The threat actor sends a request to any legitimate IRM page with a special POST parameter

2 When XModule detects this parameter, it extracts detailed instructions

3 If the instruction is "DMP", retrieve all previously exfiltrated files

4 If the instruction is "CMP," utilize a named pipe to execute a command through the Micro Backdoor component and then return the results

5 Return the results in a response body (encrypted)

Malicious IIS Native Module - XModule

We've seen the malicious module in various stages of development, with some functionalities being partially implemented or entirely missing in some samples, which shows that this "project" was either under development or specially crafted for each victim.

The samples represent an IIS native module that implements a handler for OnSendResponse and registers it with medium priority. The main functionality of the handler is to log sensitive data (passwords, card information), but it can receive various commands and it can also send commands to the MicroBackdoor process using a named pipes, acting as a proxy between MicroBackdoor and the C2 - so the MicroBackdoor process doesn't directly communicate with the C2, being harder to detect the malicious traffic.

Depending on the request method, XModule does this:

↳ if the request is GET:

- if the URL contains the "/irmng/polyfills-es2015." string (this being a .js file used by IRMng booking engine), the module will inject a malicious javascript that will set a cookie named "__gglmap" with information about the card holder and the card used to make the booking
- the following information is collected: firstname, lastname, address1, address2, city, state, country, postal code, email, card holder, card number, CVV, expiration month & year, billing zip and other data identified by elements with the following ids (that we couldn't identify): "tc91", "tc90", "tc92", "tc93", "tc94", "tc95"
- if the cookie named "__gglmap=" appears in the headers, the module logs its value (encrypted using a simple xor) in a file named "C:\\ProgramData\\Actian\\Cache\\Logs\\error_<year><month><day>_<crc32_of_cookie_value>.dmp"

↳ if the request is POST:

- if the request body contains one of the strings "password", "pass", "pwd" (so it contains a password), the body is written (encrypted) into a file named "C:\\ProgramData\\Actian\\Cache\\Logs\\info_<year><month><day>_<crc32_of_body>.dmp"

- if the request body contains one of the strings "cvv", "cvc", "cardnumber", "cardholder", "ccnum", "cc_num", "i4g0", "tenerum" (so it contains card information), the body is written (encrypted) in a file named "C:\\ProgramData\\Actian\\Cache\\Logs\\dump_<year><month><day>_<crc32_of_body>.dmp"
- if the request body contains "Module=BookData&Booking_ID=" and "&Booking_Info=", then the malware receives a command as the value of the "Booking_Info" POST parameter; the command is base64 encoded and encrypted with a simple XOR and can be one of <PIN|INF|CMD|CMP|DMP>|[optional_data]; the module executes the command handler and sends the result in the response body (encrypted):

1. PIN: the module responds with "PONG" (in the response body)
2. INF: the module responds with a small fingerprint (in the responds body): <username>|<computer_name>|<user_is_admin>|<integrity_level>
3. CMD: the module receives a command that is executed using "cmd.exe /c" (or %COMSPEC%) and sends the result as the response body
4. CMP: the module receives a command for the MicroBackdoor process that it will be written to the pipe used by MicroBackdoor (\\\\.\\pipe\\xrpcxdsvc); the command result is read from the pipe and copied to the response body
5. DMP: the module appends the content of all the files in "C:\\ProgramData\\Actian\\Cache\\Logs" folder to the response body, so

   this is how the actor exfiltrates the collected data

IOCs

XModule

filepath	md5
%PROFILES%\\<PROFILE>\\desktop\\urlmodz.dll	cb911c01d89b3a35bb3a7f525021b609
c:\\temp\\test_regmodule.dll	504a54e53727d418003d7b71647f6230
c:\\temp\\test_regmodule.dll	87397cdbf0e62dd422dfcd0c54b39710
c:\\temp\\test_regmodule.dll	07c7dffc9237373eeca170cc332e5ee0
c:\\temp\\test_regmodule.dll	5955f14160bb8aacc63f620065317c2c
c:\\temp\\test_regmodule.dll	3411c97b2017c5a60bacbae722afa9e3
c:\\temp\\test_regmodule.dll	58e200a60c8329058bb7e63118e6ce3f
c:\\temp\\test_regmodule.dll	225fee186de514e1a24536a95bfa118d
c:\\temp\\test_regmodule.dll	1d6554842dc48fd87b83113318e9256d
c:\\temp\\test_regmodule.dll	f74271e58d20f42be4bf2a685c78217d
c:\\temp\\test_regmodule.dll	a1d80427445b6db77daa39dfb89d3c2f
c:\\temp\\test_regmodule.dll	d98ef3e72ed8cac642a51498ff67c3b0

filepath	md5
c:\\temp\\test_regmodule.dll	ce7fde78cb3d3fa08e053d8a6ccdb931
c:\\temp\\test_regmodule.upx.dll	284e2bef6bec53942442a80daa3ab56d
c:\\temp\\test_regmodule.dll	91f0ba3999e7d368b294f8dd2b326865
c:\\temp\\test_regmodule.dll	3ee42bc3f765c8ac7e0708641fab4e9e
c:\\temp\\test_regmodule.dll	063588bfda9642c835b5a2bcfedaf1da
c:\\temp\\xmodule.dll	ac7cdd4d1d08f74a7f9c56b760aa991b
c:\\temp\\xmodule.dll	b487e7bfd88aa57ccbf47568055f47da
c:\\temp\\xmodule.dll	a92e0651bc8fef306ecbe992351d311f
c:\\temp\\xmodule.dll	adecf847a06fb12589e92c522f59473f
c:\\temp\\xmodule.dll	800294f84f61830b79577d241ef6c7df
c:\\temp\\xmodule.dll	8f39d5df4c38c2b90e5b9b091458eed7
c:\\temp\\xmodule.dll	61cac5c66aaba9f896da026bb2a7c899
c:\\temp\\xmodule.dll	c453f876e25c8a04d9cde58f9290c88f
c:\\temp\\xmodule.dll	b1703ed5441ac3fb5004a37722e14b22
c:\\temp\\xmodule.dll	89c283604857ea44ef8d0bc109d53a73
c:\\temp\\xm32.dll	3b15d7a3e9eea9c403ddc4e74fd329aa
c:\\temp\\xmodule.dll	450d4c982490350082ca3dc89a0e8ee2
c:\\temp\\xmodule.dll	05b0418263ac7ab3431f2329d0d3e2b0
c:\\temp\\xmodule.dll	bdbeb4dd064cc30a3c02cfb1ea0e5dc8
c:\\temp\\zxmod33.dll	366f6e5b7db3c0ef0eaa8776ae7ade24
c:\\temp\\xmodule.dll	953ccf89d1452a7142a1d3970219ed8a
c:\\temp\\xmodule.dll	c0ba71922ba520ad479f4b77d6e70688
c:\\temp\\xmodule32.dll	871de9bf5a4fdfa5e448f47a14259335
c:\\temp\\xmodule.dll	adcc2d68a2d7c5e830be550890efd42b
c:\\windows\\system32\\inetsrv\\issrpch64.dll	d5373e33861c09697af6c62987983321
c:\\windows\\system32\\inetsrv\\issrpch.dll	24d7baab665b51719aca24718e3d0115
c:\\windows\\syswow64\\inetsrv\\urlmodz.dll	cb911c01d89b3a35bb3a7f525021b609

Micro Backdoor

filepath	md5
c:\\windows\\syswow64\\logapi64.dll	e919e2ca19daa40904000a3222963b21
c:\\windows\\system32\\logapi64.dll	e919e2ca19daa40904000a3222963b21

Micro Backdoor Installer

filepath	md5
C:\\Windows\\System32\\wow64logf.dll	6d85ea5b1d88aadd43fec8a53662c0ad
C:\\Windows\\System32\\wow64log.dll	6d85ea5b1d88aadd43fec8a53662c0ad

Installer DLL

filepath	md5
c:\\windows\\system32\\batchd.dll	12f2a5faa01efcee7a0829133173da2b

Installer .bat

filepath	md5
c:\\irmsetup\\install.bat	fc45969de0677b995bfbc829906871f5

Privilege escalation & defense evasion

filepath	md5
c:\\temp\\tmp\\uninstall_2.exe	bfea2b4a02a8044cb5f7fccc36172460
c:\\temp\\tmp\\procghost.exe	4912f690fc30bb2217d1b1f3029003fe
c:\\temp\\tmp\\bitsarbitraryfilemoveexploit.exe	582862be0c3bdda4f65376169c57af98

Webshells

filepath	md5
c:\\inetpub\\wwwroot\\rdprepository\\irm\\content\\<SERVER>\\<RESORT>\\eval_full_fud.aspx	7efc7f94cbbc3e1d38873039996efe64
c:\\inetpub\\wwwroot\\rdprepository\\irm\\content\\<SERVER>\\<RESORT>\\eval_full_fud.aspx	714f7493b7eb384f3ef7a49b73f8c66f
c:\\inetpub\\wwwroot\\rdprepository\\irm\\content\\index.aspx	9cf1bbd0d83d5701aebdba6e05f7bb93
c:\\inetpub\\wwwroot\\irmcms\\custom\\31pip2pi.m3i	9cf1bbd0d83d5701aebdba6e05f7bb93
c:\\inetpub\\wwwroot\\irmcms\\custom\\pcnlgjs1.rcc	9cf1bbd0d83d5701aebdba6e05f7bb93
c:\\inetpub\\wwwroot\\rdprepository\\irm\\content\\<SERVER>\\<RESORT>\\index.aspx	45ff3ba7c1ebc1db28d4438691b13bea
c:\\programdata\\xrm\\data\\i.dat	45ff3ba7c1ebc1db28d4438691b13bea

Custom PSQL Tool

filepath	md5
C:\\ProgramData\\xrm\\files\\consoleapplication5.exe	5db5a373b1395d9f6aeb87f99e8a801c

Network

telecomptd[.]org

Romania HQ
Orhideea Towers
15A Orhideelor Road,
6th District,
Bucharest 060071
T: +40 21 4412452
F: +40 21 4412453

US HQ
3945 Freedom Circle,
Suite 500, Santa Clara,
CA, 95054

bitdefender.com