



Richkware

Framework per la creazione di malware per Windows

Riccardo Melioli

2017

# Obiettivo

## Obiettivo

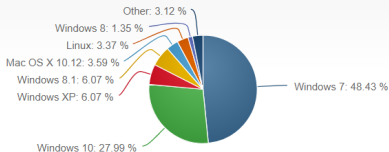
Creazione di una libreria, a scopo didattico, che permetta lo sviluppo di qualsiasi tipo di malware, in modo versatile e semplice.

## Sistema operativo target

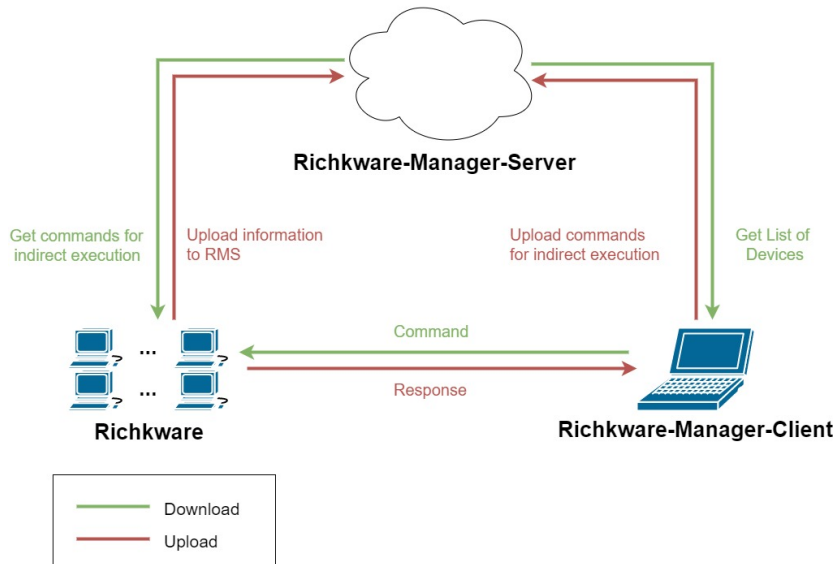
Il progetto è stato sviluppato per il sistema operativo **Microsoft Windows** come obiettivo degli attacchi, questo per motivi di **quantità di vulnerabilità** solitamente presenti in questo sistema durante l'anno, che potrebbero essere sfruttate per ottenere maggiori funzionalità. Inoltre Microsoft Windows risulta essere il sistema operativo **più diffuso** attualmente, permettendo perciò al malware di raggiungere più persone possibili.

## Desktop Operating System Market Share

August, 2017



OPERATING SYSTEM	TOTAL MARKET SHARE
Windows 7	48.43%
Windows 10	27.99%
Windows XP	6.07%
Windows 8.1	6.07%
Mac OS X 10.12	3.59%
Linux	3.37%
Windows 8	1.35%
Mac OS X 10.11	1.09%
Mac OS X 10.10	0.72%
Windows Vista	0.46%
Windows NT	0.31%
Mac OS X 10.9	0.26%
Mac OS X 10.6	0.09%
Mac OS X 10.8	0.09%
Mac OS X 10.7	0.07%
Mac OS X 10.13	0.02%
Mac OS X 10.5	0.01%
Windows 2000	0.00%
Mac OS X 10.4	0.00%



## Cos'è

Libreria di funzioni, relative alla sicurezza del sistema operativo e delle reti, utilizzabili per la creazione di un applicativo malevolo(malware). La composizioni di tali funzioni, secondo diverse logiche permettono all'applicativo di poter assumere comportamenti associabili ai seguenti tipi di malware:


- Virus
- Worm
- Bot
- Spyware
- Keylogger
- Scareware

## Cos'è

Servizio web per la gestione degli host, cioè le varie istanze di Richkware presenti. Memorizza in un database SQL tutte le informazioni relative al malware:

- **Name:** nome del dispositivo in cui è presente il malware
- **IP:** indirizzo IP da cui si connette il malware, questo dato è particolarmente utile perché fornisce dati sulla connessione utilizzata.
- **Server Port:** porta TCP aperta per permettere la connessione da remoto e per poter eseguire comandi o altre funzionalità sulla macchina infetta
- **Last Connection:** ultima data e ora in cui il malware ha contattato il server.
- **Encryption Key:** chiave di crittografia generata lato server, con cui il malware critterà dati e canali di comunicazione.

## List of Devices

Name	IP	Server Port	Last Connection	Encryption Key		
k	192.168.99.1	none	2017.09.04.11.27.50	uMVBjDfAG8DPRGYA6F8cm7O8S4oTj3Lp	<a href="#">Edit</a>	<a href="#">Remove</a>
RICHK/Richk	192.168.99.1	6000	2017.09.05.13.27.44	AupMwD0fXbJC5hk1WzNih3ClzmUjUaDA	<a href="#">Edit</a>	<a href="#">Remove</a>
y	192.168.99.1	none	2017.09.04.11.27.57	cOe7ABocPRDR7odxPdEHly4VJe2JJhIP	<a href="#">Edit</a>	<a href="#">Remove</a>
yo	192.168.99.1	none	2017.09.04.11.28.01	yrTQfscJxv4s2dn7uxVAsSbwElqxW3D6	<a href="#">Edit</a>	<a href="#">Remove</a>
yop	192.168.99.1	none	2017.09.04.11.28.09	Mrbmall39psUHFfsJ6tmuZnAuesPr2an5	<a href="#">Edit</a>	<a href="#">Remove</a>
yopo	192.168.99.1	none	2017.09.04.11.28.21	MqswVbe1idUoxy2RF0GFwnLLCDvh6BV6	<a href="#">Edit</a>	<a href="#">Remove</a>
yopoi	192.168.99.1	none	2017.09.04.11.28.26	oZgVGRCIZuHVWVA4xOPyQtQhglwb3a1O	<a href="#">Edit</a>	<a href="#">Remove</a>
yopolji	192.168.99.1	none	2017.09.04.11.28.43	gmCMCxmFlJaaCUqRWVyh1QsE3ugX4ILU	<a href="#">Edit</a>	<a href="#">Remove</a>
yopoljiji	192.168.99.1	none	2017.09.04.11.28.47	vGkQARMU0iNICDhN5NRWj1QXRimbfbmw4	<a href="#">Edit</a>	<a href="#">Remove</a>



# RMC - Richkware-Manager-Client

## Cos'è

Client di Richkware-Manager-Server, ottiene la lista di tutti gli host dal server e permette di inviare comandi da eseguire sulla macchina infetta mediante canale sicuro.

# RMC - Richkware-Manager-Client

Richkware-Manager-Client

File Edit View Help

http://192.168.99.100:8080/Richkware-Manager-Server/DevicesListAJAJ

Connect ☐ Encryption (RMS) Disconnect

Name	IP	Server Port	Last Connection	Encryption Key
k	192.168.99.1	none	2017.09.04.11.27.50	uMVBjDfFaG8DPRGYA6F8cm708S4oTj3lp
RICHK/Richk	192.168.99.1	6000	2017.09.05.21.34.17	AupMwD0fxbJC5fk1Wznlh3ClzmUjJuaDA
y	192.168.99.1	none	2017.09.04.11.27.57	cOe7ABocPRDR7odxPdEHiy4VJe2JJHP
yo	192.168.99.1	none	2017.09.04.11.28.01	yrTQfscJxv4s2dn7uxVAsSbwElqxW3D6
yop	192.168.99.1	none	2017.09.04.11.28.09	MrbmaII39psUHFsj6tmuZnAuesPr2an5
yopo	192.168.99.1	none	2017.09.04.11.28.21	MqswVbeIidUoxy2RF0GFwnLLCDvh68V6
yopoi	192.168.99.1	none	2017.09.04.11.28.26	oZgVGRClZuHVVWA4xOPyQtqhgIwb3a1O
yopoji	192.168.99.1	none	2017.09.04.11.28.43	gmCMCxmfTjaaCuqRWVvYH1QsE3ugX4ILU
yopojji	192.168.99.1	none	2017.09.04.11.28.47	vGkQARMU0NICDhNSNRWj1QXRimbfbmw4
yopojjji	192.168.99.1	none	2017.09.04.11.28.51	xZyWbj3JMaEPxb69ICaIlkzip59TZuav
yopojjjji	192.168.99.1	none	2017.09.04.11.28.55	bx26G4dULNOaBFsmEF4KfURodL4gSd4

☐ Direct ☐ Force Encryption (Richkware) 192.168.99.1:6000 ls

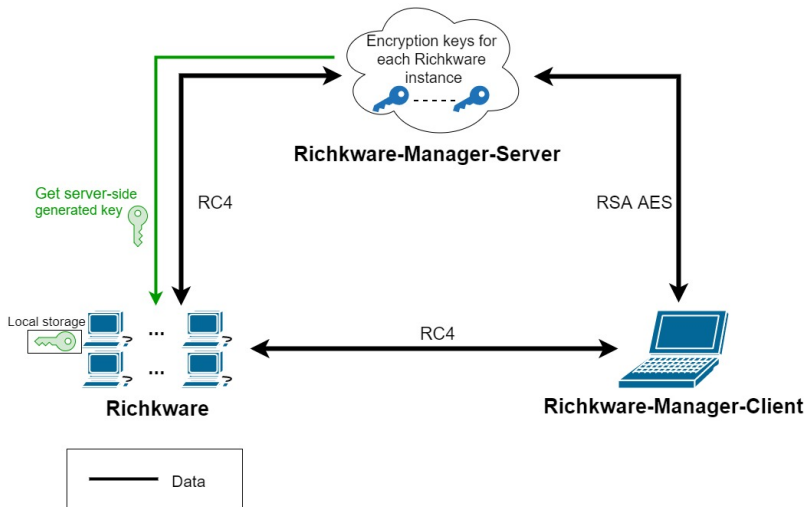
Connect Device Disconnect Device Send Command

```
ls
-->
CMakeCache.txt
CMakeFiles
Makefile
Richkware.cbp
Richkware.exe
cmake_install.cmake
```

## Crittografia in Richkware

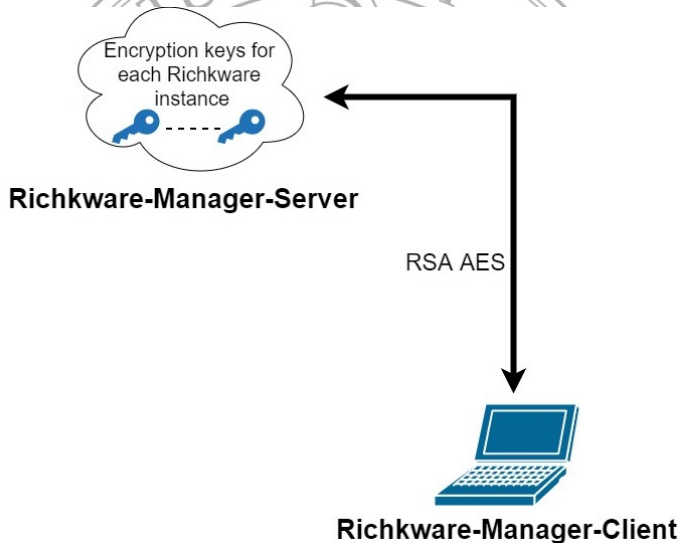
Utilizzata nella protezione dei **canali di comunicazione** tra Richkware, RMC e RMS, ma anche per la protezione dei **dati conservati** su file o voci di registro da parte di Richkware, infatti qualsiasi informazione salvata nel sistema dall'istanza di Richkware viene crittata.

## Cryptography



## Fasi

- ① RMC genera la sua coppia di chiavi RSA
- ② RMC invia la sua **chiave pubblica RSA** a RMS
- ③ RMS
  - ① genera la sua coppia di chiavi RSA e la chiave AES che verrà utilizzata per crittare i messaggi.
  - ② firma la chiave AES con la sua chiave privata
  - ③ critta con la chiave pubblica di RMC il pacchetto formato dalla firma e il messaggio (Chiave AES)
  - ④ invia a RMC il pacchetto formato da: **chiave pubblica di RMS**, e **pacchetto crittato contenente la firma e messaggio**.
- ④ RMC decritta con la sua privata e verifica il contenuto con la chiave pubblica di RMS.
- ⑤ RMC può utilizzare la chiave AES per decifrare i dati successivamente ricevuti



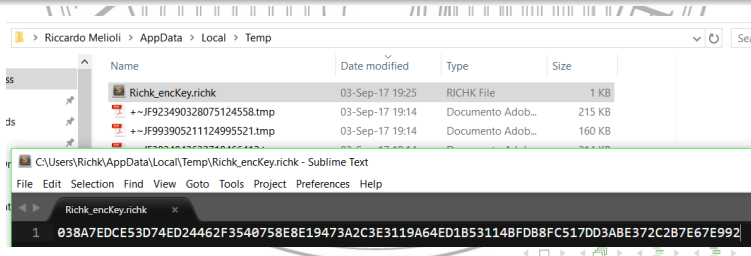
# Pacchetti Scambiati

```
GET /Richkware-Manager-Server/DevicesListAJAJ?  
encryption=true&Kpub=3082012230.....03010001 HTTP/1.1  
User-Agent: Java/1.8.0_91  
Host: rms-richk.rhcloud.com  
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2  
Connection: keep-alive
```

```
HTTP/1.1 200 OK  
Date: Sun, 03 Sep 2017 16:29:25 GMT  
Server: Apache-Coyote/1.1  
Set-Cookie: JSESSIONID=1F479C58D327620765A0D431A15DCB84; Path  
=/  
Richkware-Manager-Server/; HttpOnly  
Keep-Alive: timeout=15, max=100  
Connection: Keep-Alive  
Transfer-Encoding: chunked  
Content-Type: text/plain  
{'encryptedAESsecretKey' : '8F01D550529.....C2205EC',  
  'signatureAESsecretKey' : '496D636344.....773D3D',  
  'kpubServer' : '3082012.....03010001',  
  'data' : '33A07E.....BFDCEC'}
```

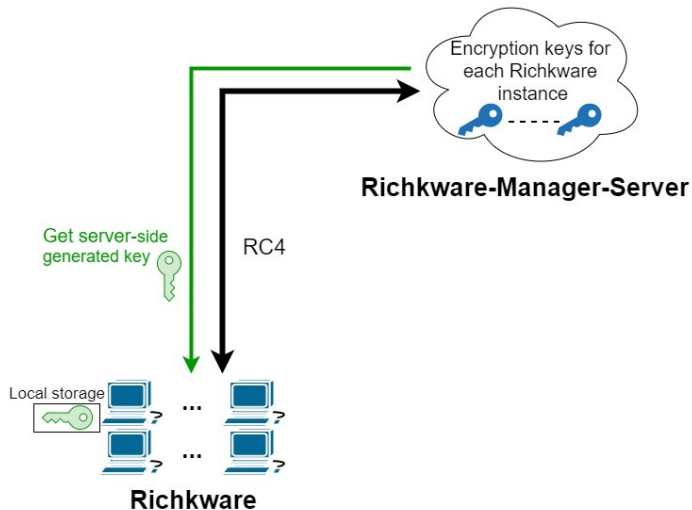
# Crittografia RMS - Richkware

Per la comunicazione tra Richkware e RMS, si utilizza l'algoritmo RC4, particolarmente veloce e performante. Al primo avvio Richkware utilizza una **chiave hardcoded** al suo interno, appositamente scelta dal creatore dell'istanza di Richkware, quindi presente anche sul server RMS, dopo di che, quando verrà stabilito il contatto tra Richkware e RMS, l'istanza otterrà la chiave di crittografia dal server e utilizzerà sempre quella, salvandola localmente in un file crittato, questo per rendere più sicuro il funzionamento ed evitare attacchi di tipo **disassembling** fatti all'eseguibile dell'istanza di Richkware.





# Crittografia RMS - Richkware



## Crittografia RMS - Richkware

Per la comunicazione tra Richkware e RMC, si utilizza l'algoritmo RC4. Vi è un **protocollo** di comunicazione alla base della comunicazione tra RMC e Richkware, la crittografia si applica ad ogni pacchetto scambiato nel protocollo, tranne che all'**handshake** iniziale, appunto per sapere se Richkware supporti o meno la crittografia. Se il canale è insicuro, quindi un utente malevolo volesse inserirsi nella comunicazione e cambiare il messaggio di handshake da canale crittografato a non crittografato, si può impedire tale attacco abilitando la modalità del client "**Force Encryption**" che permette al client di ignorare l'handshake e procedere con la crittografia in ogni caso. Per utilizzare questa modalità in RMC, il server in Richkware deve aver abilitato la crittografia.

# Crittografia RMC - Richkware



## Protocollo di comunicazione

Il protocollo permette all'utilizzatore di RMC di poter interagire con la macchina dov'è installata l'istanza di Richkware.

Le richieste ricevute dal server in Richkware, presente nella classe Server vengono mandate ad un Dispatcher, che smista la richiesta in base a un certo codice, esegue la richiesta e ritorna la risposta, in modo che il server possa comunicarla al client connesso. Il dispatcher è implementato nel seguente modo:

```
... rimozione delimitatori

switch (commandID) {
    case 0:
        response = "***quit***";
        break;
    case 1:
        response = CodeExecution(command);
        break;
    case 2:
        //...
        break;
    default:
        response = "error: Command ID not found\n";
}
```

# Sintassi di una richiesta

## Sintassi di una richiesta

La sintassi di una richiesta è la seguente:

**[[1]]ls**

Il comando precedente, avendo il parametro 1, significa che si richiede l'esecuzione della stringa che segue come un comando da shell, quindi "ls", verrà eseguito dalla shell di Windows e la risposta, cioè quello che verrebbe stampato sulla shell in seguito a quella richiesta viene mandato al client.

Fine

**Grazie per l'attenzione**