Journal of Engineering and Applied Sciences 12 (23): 7154-7158, 2017
ISSN: 1816-949X

Hardware Firewall Bypass (HWFWBYPASS) Attack on pfSense

[1]Moran Edgar, [2]Salcedo Parra Octavio Jose and [3]Sanchez Cespedes Juan Manuel
[1]Department of Systems Engineering, Faculty of Engineering,
Nacional University of Colombia, Bogota D.C., Colombia
[2]Department of Computer Engineering,
Francisco Jose de Caldas District University, Bogota D.C., Colombia
[3]Department of Electronic Engineering, GIIRA Investigation Group, Faculty of Engineering,
Francisco Jose de Caldas District University, Bogota D.C., Colombia

Abstract: This study provides documentary evidence of the evaluation and implementation of pfSense's defense and protection mechanisms to avoid a HWFTBYPASS (Hardware firewall bypass) attack which is implemented through Remote Desktop Protocol (RDP). pfSense is an extremely robust open network security software. To achieve the above, the part of the attack that corresponds to the bypass the hardware firewall will be implemented in such a way that it will be possible to infer whether pfSense is by default vulnerable or not and in that case, mitigation methods will be addressed.

Key words: pfSense, remote desktop protocol, hardware firewall by pass, attack, sense, mitigation

INTRODUCTION

At DevCom DEF CON 22 (Balazs, 2014) presented the "shiny tool", capable of performing a HWFWBYPASS (Hardware firewall bypass) attack through RDP (Remote Desktop Protocol).

This attack consists of taking command and control of a workstation. To achieve this, the attacker will wait until a legitimate user connects to the Windows RDP server in a secure way, starts a session on this computer and once this is done, the following steps are performed:

- A malware is installed on the RDP server
- Any code is executed on the RDP server
- Logged user escalates its privileges to administrator privileges on the RDP server
- Firewall bypass is performed

From the previous four steps, only step 1 and 4 are related to the new tool presented at the conference that was developed by Balazs (2014) Apart from that, the RDP being protected by a firewall uses a two factor authentication system (Karapanos *et al*., 2015) that doesn't allow files to be directly copied between the client and RDP server and has a white list policy implemented by an application called applocker which resides on the RDP server.

This document presents an evaluation performed on the pfSense firewall in order to execute the four-step process of the HWFWBYPASS attack and in this way to check the state-of-the-art pfSense firewall. In the case that it is vulnerable by default, a mechanism to detect and mitigate this attack will be implemented.

RDP is a widely used software, mostly because it allows one to connect to remote Windows servers in a simple and fast way which also makes it very attractive to attackers who attempt to take control of servers on local area networks of companies and homes in order to perform different types of attacks.

MATERIALS AND METHODS

Not much information on precedents to HWFWBYPASS attack through the RDP service was found in related literature. Even though there are other types of attacks that are done to RDP servers such as DoS (Denial of Service) attacks reported in the Microsoft Security Bulletin (SANS Institute, 2007). Specifically, expedient MS01-006 reports that badly formed data cause the RDP server to stop responding, expedient MS02-051 documents that a malicious packet could restart the service and could give access to the information that goes through the RDP connection. Some of these attacks are the forefathers of other attacks of the remote exploit type (Whitman and Mattord, 2011) which allow attackers to take control of a machine (Peikari and Chuvakin, 2004) and of the installed Malware Technet Microsoft.

Corresponding Author: Moran Edgar, Department of Systems, Engineering, Faculty of Engineering,
Nacional University of Colombia, Bogota D.C., Colombia

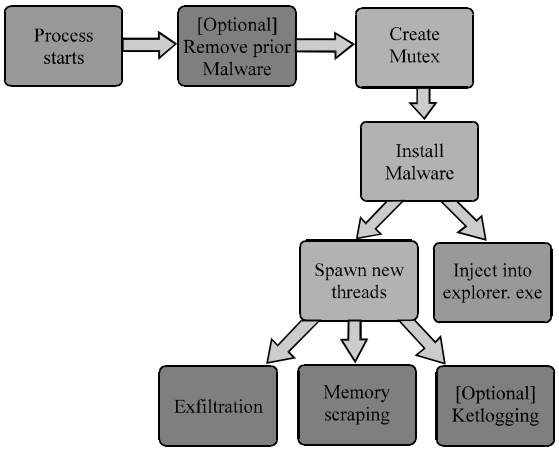

Fig. 1: Alert (TA14-2124[a])

On July 31st, 2014 the Computer Emergency Readiness Team at Department of Homeland Security of United States of America, generated an alert (TA14-212) (US-CERT., 2014) which showed a new family of malware called "backoff" whose capabilities are: memory scanning, scraping memory for track data, logging keystrokes, command and control (C2) communication and injecting malicious stubs onto explorer.exe files (Fig. 1).

Even though the alert (TA14-212a) doesn't talk about the HWFTBYPASS attack, a resemblance to the alert's malware can be found in that both take control of the machine. Nevertheless, the difference is that the transmission of information is done mainly via http to other servers and not by redirecting ports in order to open a communication channel to transfer the information as the HWFTBYPASS attack does (US-CERT., 2014).

The proposed mitigation to alert TA14-212a is the use of a firewall (US-CERT., 2014). Taking into account that this alert doesn't take into account the HWFTBYPASS type of attack, it doesn't provide a direct way to prevent this attack. Thus, the ports of incoming connections should be analyzed and controlled proactively because this is the only way to detect and avoid this attack.

The proposed methodology that will be followed to achieve the goal of this study is described in five phases.

Clarifications: Taking into account that the main goal of this research is to analyze the fourth step of the attack (the firewall bypass) the proposed scenery which varies slightly from the original is the following: given a RDP server with just one authentication factor (local user and password) and the first 3 steps of the attack assumed as granted, the goal of this study is to show how HWFTBYPASS could be avoided and how a mechanism to mitigate this type of attack can be implemented on pfsense.

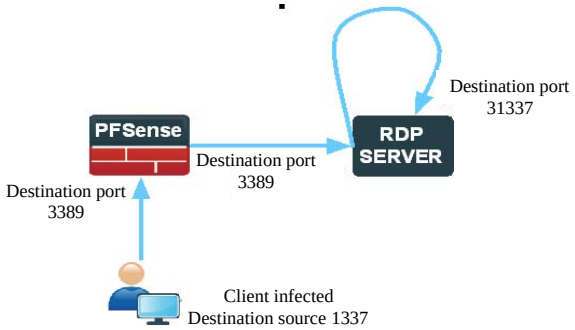

Fig. 2: Configuration (generated with omni Graffle)

Test environment setup: The fourth th step of the attack will be performed on the following test environment. This environment will be installed on a hypervisor with three virtual machines. The following are the general steps to setup the test environment:

- Install "shiny tool" on the RDP server to accomplish the 4 th step and netcat in order to transmit information to the client machine
- Install netcat on the RDP client to accomplish the data transmission to the RDP server
- Setup a redirection rule of port 3389 to the RPD server on the pfSense firewall in order to establish the connection between the client and the RDP server

Attack execution and pfSense vulnerability assessment: The fourth step of the HWFWBYPASS attack is performed and pfSense (installed by default) is assessed in order to determine if it could avoid the attack.

Attack assessment: The analysis of the attack is performed.

Implementation of the defense or protection mechanism: In case of pfSense presenting vulnerability, a defense to avoid or mitigate this HWFWBYPASS attack will be implemented.

RESULTS AND DISCUSSION

Test environment setup: For the test environment, the following configuration was setup (Fig. 2).

Virtual environment: VMware Fusion 8.0.2 installation on a Mac Book Pro (late 2012 8 GB RAM and 2.5 GHz Intel Core i5 processor) with Mac OS X 10.11.1 operating system.

Client virtual machine: For the client workstation (Fig. 1) Windows 8 was installed with one processor and 1 GB RAM memory. To accomplish the fourth step, netcat was installed through Nmap.

RDP server virtual machine: For the client (Fig. 1) Windows 8 was installed with one processor and 1 GB RAM memory. To accomplish the fourth step, netcat was installed through nmap.

PfSense virtual machine: pfSense Version 2.4.4 was installed and also a new redirection rule was added to the WAN interface, consisting of redirecting port 8839-8839 of the RDP server and two more rules were added in order to access the firewall to be administered via the web on port 7880 and via ssh on port 22.

Attack execution and pfSense's vulnerability assessment: On the server, 2 command terminals should be open with administration privileges. Then, the following commands should be executed on each terminal (Fig. 3).

The 1st command opens a connection to port 31337 and the second executes shiny tool (developed to performed the fourth step of the attack). Shiny tool redirects all traffic that gets to port 3389 coming from port 1337- 31337 where ncat established the connection.

On the RDP client machine, the connection with the RDP server machine must be established first, then a terminal with administrator privileges must be open and the following command must be entered C:\> ncat-p 1337 192.168.2.47 3389.

On the RDP client machine, a new command terminal must be opened with administrator privileges and the following command must be executed to establish the connection with HWFWBYPASS.

To assess whether the HWFWBYPASS attack could or could not be performed, the following command must be entered on the command terminal of the client machine (where ncat was executed): HELLO attack.

Then, the command terminal the on RDP server should be checked (taking into account that it is a Windows 8 machine, we could access it through the RDP connection that has already been established). Where the ncat command was executed it can be determined whether the "HELLO attack" information has been transferred. In order to test whether the server is able to send information, the text "SYN" can be entered on the command terminal (just mentioned) and it can be seen that it is effectively transmitted to the client (Fig. 4).

Attack assessment: During the attack process, it was noticed that PFSsense (installed by default) could not mitigate or avoid the attack. Furthermore, the analysis of the data packets was performed with tcpdump and it was found that when the client connects to the RDP server, it opens only one random port. During the tests when the attack starts there is one more connection established

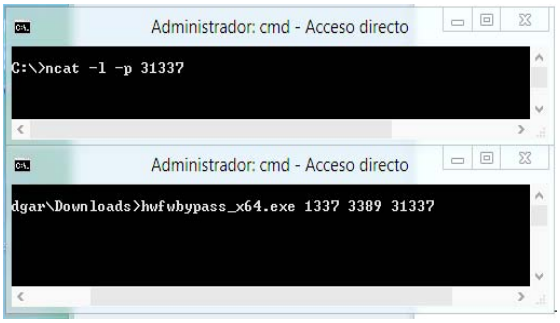

Fig. 3: Commands (generated via screenshot of CMD)

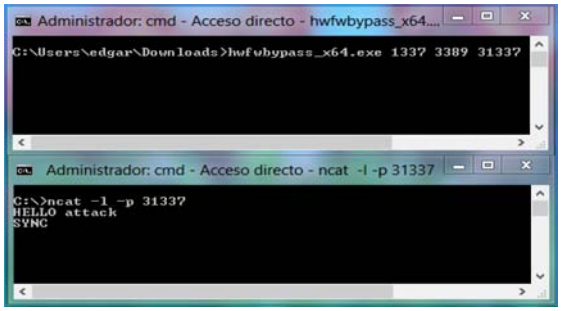

Fig. 4: HELLO attack (generated via screenshot of CMD)

(in this case, it is port 1337). Thus, with this observation it can be understood that RDP opens only one source port and that when the attack begins there is one more source port open. This can be used to identify the attack on pfSense following a traffic analysis. At the moment of detecting a connection from two different source ports, it can be inferred that the attack is happening.

Implementation of the defense or protection mechanism: As the attack was effectively performed and pfSense could not mitigate it, a new development using php and tcpdump is proposed for integration into the pfSense firewall so it will execute the following process: traffic coming from the entrance (WAN) interface port 3389 will be captured using tcpdump and will be sent to a php script with the following command line: "tcpdump -i em0 dst port 3389 | php test1.php".

The PHP script will process the information and will determine if two ports are being opened from the same client IP. If that is the case, it will send a blocking rule to the pfSense firewall and then it will flush all actual connections so it will drop the RDP connection from the infected workstation. The following is the source code of the mitigation script written in PHP (Fig. 5-7). Even though there are several ways to mitigate attacks to RDP servers using security technologies such as authentication with kerberos, SSL or VPN (Longzheng *et al*., 2004;

```
#!/usr/local/bin/php -q
<?php

/*
El siguiente programa evita el ataque HWFTBYPASS (unicamente para IPv4),
procesa la información que recive del comando tcpdump
"tcpdump -i em0 dst port 3389 " y procesa la información recibida si detecta
que desde la misma IP se abren dos puertos de origen diferente agrega una regla
de bloqueo de la IP y hace un limpieza de reglas con el fin de eliminar
la conexión del ataque.
*/
//TIEMPO EN QUE MANTIENE LA TABLA DE  DIRECCIONES IP Y PUERTOS FUENTE EN SEGUNDOS
define("TIEMPOSESTADO","120");
//identifiacion de la interface de red en pfsense

define("NETINTERFACE","WAN");


//DEFINICION DE LA ENTRADA Y SALIDA DE CONSOLA
if (! defined(STDIN)) {
        define("STDIN", fopen("php://stdin", "r"));
}
if (! defined(STDOUT)){
        define("STDOUT", fopen('php://stdout', 'w'));
}

// TABLA DE TCPDUMP DURANTE EL PERIODO TIEMPOSESTADO
$tableIPall = array();

// TABLA DE DIRECCIONES IP Y PUERTOS FUENTE

$tableIP = array();
# leer la entrada estandar

$timestart=  microtime()/1000;
$i=0;
```

Fig. 5: The code of the mitigation script 1 (generated via. screenshot of atom editor)

```
$i=0;
while( !feof(STDIN)){
        //tomar una linea de la entrda estandar
        $line = trim(fgets(STDIN));

        //procesar la linea por separador de espacio
        $data   = preg_split('/\s+/', $line);
        $tableIPall[$dat[0]] = $data;
        #print "data:".$data[0]."\n";
        list($IPv4, $port_ori)=explode(":", get_IPv4_port($data[2]));
        //si la IP esta en el arreglo
        if (find_in_matrix($tableIP , $IPv4)) {
            if(!find_in_matrix($tableIP , $port_ori)){
                //bloquear el acceso
                print "\nBLOQUEADO\n easyrule block ".NETINTERFACE." ".$IPv4;
                print "pfctl -F all";
                exec("easyrule block ".NETINTERFACE." ".$IPv4);
                exec("pfctl -F all");
            }
            else {
              $tableIP[$i][0]=$IPv4;
              $tableIP[$i][1]=$port_ori;
              $i++;
            }
            //revisar el puerto
        }
        else {
            //Registro unicamente de IPv4
            $tableIP[$i][0]=$IPv4;
            $tableIP[$i][1]=$port_ori;
```

Fig. 6: The code of the mitigation script 2 (generated via. screenshot of atom editor)

Huang *et al*., 2009; Rouzaud and Viot, 2007) these mechanisms will not foresee these types of scenarios nor will they be able to avoid a HWFTBYPASS attack. This is because if the operating system of the server or the software has any security issues that allow the escalation of privileges through any type of attack then a HWFTBYPASS attack could be performed. So, in order to mitigate this attack, it is very important to use highly reliable and secure operating systems, that implement a policy regarding risk mitigation and that include some characteristics such as memory encryption, RNG, omalloc, separation of privileges, chroot and pledge, among others. The former could avoid most attacks through the proper execution of the operating system which highly differs from an ecosystem full of bugs.

There are some measures that can be taken to audit RDP which could verify whether the attack is happening and could offer a better solution than that proposed in this study, at the level of the pfSense firewall. By having a dedicated specific machine acting as a proxy server (Fig. 8), it could only analyze incoming traffic and in the case of having some other services that were remotely accessed on the RDP server such as a http server, the proxy server would not give a "false positive". On the other hand in the case of the proposed solution in the former section, it will suffice only if the RDP service is the

```
            $tableIP[$i][1]=$port_ori;

            $i++;
        }

        //vaciar la tabla de direcciones IP si ha trascurrido el tiempo de verificación
        $timenow=microtime()/1000;
        if (($timenow-$itmestart)> TIEMPOESTADOS)
            empty($tableIP);
}
/*
Busca un elemento en la matriz si existe retorna verdadero
en caso contrario retorna falso
*/
function find_in_matrix($matriz,$data) {
        if ( empty( $matriz )){
            return false;
        }
        for ($i=0;$i<count($matriz);$i++){
            if(in_array($data,$matriz[$i]))
                return true;
        }
        return false;
}
/*
de un registro con la forma xxx.yyy.zzz.www.abcd
retorna la IP y el puerto de la siguiente forma
xxx.yyy.zzz.www:abcd
*/
function get_IPv4_port($data){
      $split = explode(".", $data);
      $port_ori=$split[count($split)-1];
      $IPv4=$split[0].".".$split[1].".".$split[2].".".$split[3];
      return $IPv4.":".$port_ori;
}
?>
```

Fig. 7: The code of the mitigation script 3 (generated via screenshot of atom editor)

only service running on the RDP server and it will also avoid incurring additional infrastructure costs related to adding a new proxy server for the RDP.

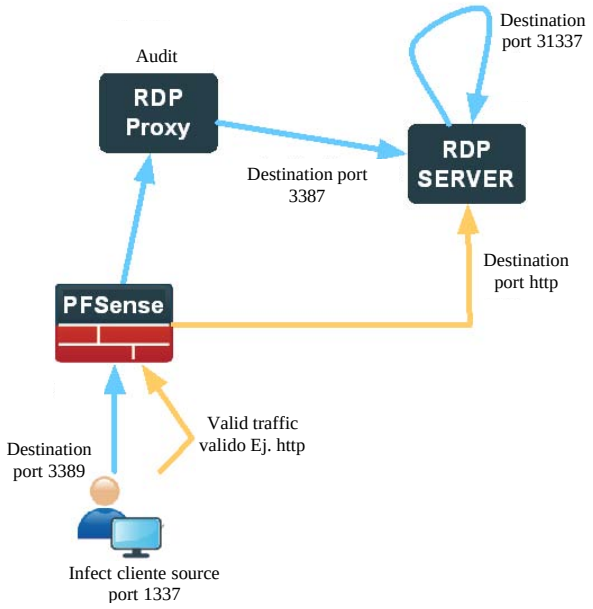

Fig. 8: Proxy server (generated via. screenshot of atom editor)

CONCLUSION

This attack is very critical and very difficult to control as the access to a workstation or RDP client which is needed to perform the attack is not complex and can be achieved through vulnerabilities in the operating system such as denial of service, exploits, malware, brute force attacks, privilege escalating software or simply by using social engineering.

Unfortunately, even though RDP is easy to implement and use, it is not advisable for critical environments or where high security requirements are essential. In these cases, neither RDP nor any direct remote access tool should be used. Nevertheless, when this type of access is a necessity there must be some additional security mechanisms such as the use of operating systems with high security at the client and the server that will not allow privilege escalation in order to avoid the attacker gaining total control of resources and information that reside on these systems.

The proposed solution for pfSense could generate "false positives" in some cases when the RDP client has access to other resources or servides other than the RDP service. This is because if other ports were open when establishing a connection for example, a web application server, it would be detected as an attack as several several ports would be open from the source and the connection would be subsequently dropped and blocked.

The only way to avoid this type of attack is to implement a mitigation policy directly on the operating system. Any other security technique would act as a patch to cover bad software or operating system design. Furthermore, non-proactive policies for mitigating these bugs should be avoided.

REFERENCES

Balazs, Z., 2014. Bypass Firewalls Application White Lists, Secure Remote Desktops in 20 Seconds. MRG Publisher, London, England,.

Huang, S.H., C. Lin, Z. Chen, X. Jiang and K. Wang *et al*., 2009. Proxy-based security audit system for remote desktop access. Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN 2009), August 3-6, 2009, IEEE, Beijing, China, ISBN:978-1-4244-4581-3, pp: 1-5.

Karapanos, N., C. Marforio, C. Soriente and S. Capkun, 2015. Sound-proof: Usable two-factor authentication based on ambient sound. Proceedings of the 24th USENIX Symposium on Security, August 12-14, 2015, USENIX Publisher, Washington, DC., USA., ISBN:978-1-931971-232, pp: 483-498.

Longzheng, C., Y. Shengsheng and Z. Jing-Li, 2004. Research and implementation of remote desktop protocol service over SSL VPN. Proceedings of the International Conference on Services Computing, September 15-18, 2004, IEEE USA., pp: 502-505.

Peikari, C. and A. Chuvakin, 2004. Securuty Warrior. O'Reilly Media, Inc., Sebastopoly, California, ISBN:0-596-00545-8,.

Rouzaud, C.J. and N. Viot, 2007. Secured architecture for remote virtual desktops. Proceedings of the 2007 international symposium on collaborative technologies and systems (CTS 2007), May 25, 2007, IEEE, France, Europe, ISBN:978-0-9785699-1-4, pp: 80-87.

SANS Institute, 2007. Windows remote desktop heroes and villains. SANS Institute, Boston, Massachusetts. https://www.sans.org/reading-room/whitepapers/windows/windows-remote-desktop-heroes-villains-2026.

US-CERT., 2014. Backoff point-of-sale malware. United States Computer Emergency Readiness Team, Washington, DC., USA. https:// www.us-cert. gov/ncas/ alerts/TA14-212A.

Whitman, M.E. and H.J. Mattord, 2011. Principles of Information Security. 4th Edn., Cengage Learning, Boston, Massachusetts, ISBN-13:978-1-111-13821-9, Pages: 623.