

MWR InfoSecurity Security
Advisory

Retain Resource Server -
Remote Code Execution

7th April 2009

MWR  INFOSECURITY

CONTENTS

1	Detailed Vulnerability Description	4
1.1	Introduction	4
1.2	Technical Background.....	4
1.3	Vulnerability Details.....	4
1.4	Exploit Information	5
1.5	Dependencies	5
2	Recommendations.....	6

Retain Server Remote Code Execution

Package Name:	Retain Resource Planner Server
Date:	2008-11-28
Affected Versions:	The vulnerability has only been confirmed in Retain Resource Planner Server version 4.2.0. However, it is expected that other, earlier versions will also be affected.
CVE Reference	Not Yet Assigned
Author	A. Plaskett
Severity	High Risk
Local/Remote	Remote
Vulnerability Class	Protocol flaw
Vendor URL	http://www.retaininternational.com/products/retainresource.htm
Version	Confirmed in version 4.2.0, although earlier versions are expected to be affected.
Vendor Response	The vendor was contacted and has provided an updated version of the Retain server. Versions 4.2.1 and 5.0 address the vulnerability identified.
Exploit Details Included	Limited information about exploit vectors is included.

Overview:

A vulnerability exists in the Retain Planner Server networking protocol which could allow an attacker to execute code remotely by crafting a malicious packet in order to hijack the flow of execution.

Impact:

The vulnerability would enable an attacker to execute arbitrary code on the system in the context of the user who executed the program (Retain Server). This vulnerability could also be used to expose confidential information stored within the application.

Cause:

The networking protocol which is used to communicate between the client and server has a fundamental protocol design issue. Memory address values are passed between the client and server in communication packets and so can be user controlled.

Interim Workaround:

The introduction of host based or network filtering controls could help to mitigate the risk from this issue. In addition, access could be restricted to authorised users only. However, this would still not prevent an authorised user from exploiting the service to gain unauthorised access to system data.

Solution:

A patch to address this issue is available directly from the vendor. Versions 4.2.1 and 5.0 address this vulnerability.



1 Detailed Vulnerability Description

1.1 Introduction

The Retain Resource Planning System is currently developed by Retain International and is described by the vendor as follows:-

“Retain Resource Planning System is a graphical scheduling software that helps lower operational costs via efficient planning, in turn giving you a competitive edge. Resource utilisation can be optimised by matching the right people to the right job. Allocation of resources can be adjusted as and when required by circumstances.”

Source: <http://www.retaininternational.com/products/retainresource.htm>

1.2 Technical Background

The Retain Resource Planning System is based on a client-server architecture. The client requests a schedule or plan from the server and the server responds with the information. Basic authentication and access rights are in place to restrict the access which individual users have to data. The networking protocol used is a typical request/response model with the first packet sent from the client being a handshake packet to which the server responds with a handshake reply. After a successful handshake an authentication sequence takes place. An authentication packet containing the username and password is sent from the client to the server. The server then replies with a response packet to notify the client of a successful or unsuccessful authentication. If the authentication is successful the client then requests scheduling or planning information from the server and the response is transmitted to the client.

1.3 Vulnerability Details

The vulnerability exists due to the way the networking protocol is implemented between the client and the server. As outlined above, the Retain server protocol is based on a typical request/response model; however, fundamental design issues in the protocol expose server memory addresses to the client. It was found that the handshake response contained a memory address (highlighted in Figure 1.0) which was returned to the client as a pointer for use in the next request (the authentication request – illustrated in Figure 1.1). The code which handles the authentication request then used this address to perform certain functionality within the application.

```

0000 30 00 00 00 06 00 00 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 18 00 00 00 02 00 00
0020 f8 c5 bb 01 00 00 00 00 00 00 00 00 00 00 00

```

Figure 1.0 – Handshake Response

```

0000 64 00 00 00 01 00 00 00 f8 c5 bb 01 00 00 00 00 d.....
0010 00 00 00 00 00 00 00 00 04 00 00 00 61 6c 65 78 .....alex
0020 03 00 00 00 06 e 6f 70 00 00 00 00 00 00 00 00 .....nop.....
0030 00 00 00 48 00 00 00 04 00 00 00 61 6c 65 78 2b ...H.....alex+
0040 b9 73 10 30 51 01 00 00 04 00 00 00 00 00 00 00 ..s.0Q.....
0050 00 00 00 00 00 0b 00 00 43 53 57 61 6c 6c 43 .....CSWallC
0060 68 61 72 74                                     hart

```

Figure 1.1 – Authentication Request

It was discovered that an attacker could craft a malicious authentication packet specifying an arbitrary memory address such that the flow of code which processed the packet data could be hijacked and used to execute code of the attacker's choice.

The severity of this issue is increased as the code which handles the attacker's specially crafted data runs before the authentication is performed and so a legitimate account is not required for exploitation.

1.4 Exploit Information

In order to exploit this vulnerability it was necessary to analyse the code which handles the address passed in the authentication packet request.

CODE:0052AA09	mov	eax, [ebp+var_14]
CODE:0052AA0C	mov	edx, [eax]
CODE:0052AA0E	call	dword ptr [edx+14h]

It was discovered that an attacker could control the value of [ebp+var_14] by specifying the value highlighted in the authentication request packet shown above; this would then be loaded into the eax register.

Memory addresses were found which pointed to an area which an attacker could control within the packet. It was possible to craft an authentication request packet which loaded an address into the eax register such that the subsequent call instruction ran the malicious code. An address was chosen such that once it was dereferenced by the next instruction and loaded into the edx register, the value of edx +14h would point to the attacker's own malicious code (typically expected to be shellcode) and execution would continue from that point after the call instruction.

The shellcode was stored in the rest of the data packet since that data was not used till later instructions. As the maximum packet size was limited it was only possible to store a small amount of shellcode in the packet. However, this was enough to include a loading step in the payload and allow a staged reverse shell to be used which would give full control to the attacker in the context of the user who executed the Retain server software initially.

The existence of this vulnerability has been confirmed by MWR InfoSecurity and working exploit code exists.

1.5 Dependencies

Exploitation of this vulnerability requires that an attacker is able to make requests to the Retain server. Therefore, network filtering can be put in place to protect a server in sensitive environments.

2 Recommendations

It is recommended that all users should install the appropriate security patch released by the vendor in response to this issue. The vendor should be contacted directly for this patch. Versions 4.2.1 and 5.0 address the vulnerability identified in this advisory.

MWR InfoSecurity
St. Clement House
1-3 Alencon Link
Basingstoke, RG21 7SB
Tel: +44 (0)1256 300920
Fax: +44 (0)1256 844083
mwrinfosecurity.com