WARCS: WIDE AREA REMOTE CONTROL SYSTEM IN SPRING-8

A.Yamashita, Y.Furukawa
SPring-8 , Sayo, Hyogo, Japan

ABSTRACT

WARCS (Wide Area Remote Control System) enables machine experts to control SPring-8 accelerator and beamlines from outside of the site. WARCS is used when a machine gets into trouble and shift crew need assistance of the experts outside of the site. A machine expert who got a emergency phone call from the shift leader can access SPring-8 control system with just two actions i.e. opening the application and entering the one time password. The expert shares control with shift crew in the control room through the internet. WARCS has been successfully working for two years. It has been tested from many places including Europe.

INTRODUCTION

SPring-8 has been maintained by small number of machine experts compared to the same size facilities. Although well trained operator crew solve daily machine problems, only machine experts can fix rare machine troubles. Operation crew give an expert emergency call when they give up fixing a trouble and the expert comes to the site. If the expert goes out of the city, telephone call was the only way to fix the trouble. Recently, experts often make trips for conference, workshop and consulting. They demand to control SPring-8 machine from outside of the site with the internet in case of emergency.

MADOCA [1] system, which controls SPring-8 accelerator complex, is a networked remote control system by nature. GUI (Graphical User Interface) applications running on workstations exchange control messages through network to the lower level control programs running on VME computers. Thus, a workstation outside of the SPring-8 site can control VME, if the workstation can communicate through the network.

Strong firewall protects SPring-8 control network from the internet. Secure tool which build ip -tunnel through the firewall enables experts control SPring-8 machines. We looked for tools in the market and finally decide to develop our own mechanism because there are no tools to meet our demands.

DESIGN

Scenario

Before we design the connection mechanism, we construct a scenario how a machine trouble is fixed.

When the shift crew give up fixing the trouble, they give an expert emergency call. Real-time machine log data of SPring-8 can be accessed via WWW [2]. The expert may solve the problem with phone and WWW. If the expert gives up fixing trouble with phone and WWW, he/she connects notebook PC to the internet. The shift leader digs a tunnel in the firewall to allow login to the workstation inside. The experts log on the workstation inside and solve the machine trouble. The shift crew can watch what the expert is doing. After the problem fixed, the shift leader break the connection and close the tunnel in the firewall.

Requirement

According to the scenario, we require the function as follows for connection tools.

1. Secure encrypted connection through the firewall.
2. Authentication mechanism.

3. Shift leader controls the connection. He/she allows connection, monitor and disconnect as needed.
4. Login with character terminal or graphical.
5. Graphical interface on the workstation can be shared with shift crew.
6. No modification to the control application.
7. Client tool is platform independent, at least Windows, Macintosh and Linux.
8. Easy operation.

SYSTEM

The outline of WARCS is shown in Fig.1.

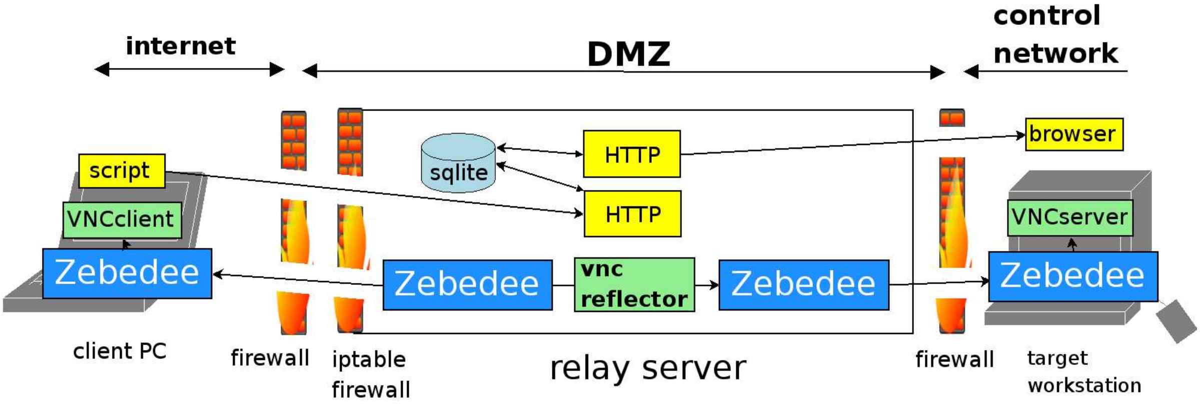

Fig.1 Outline of WARCS. The firewall between Spring-8 LAN and the internet is not drawn.

A Linux server installed in the DMZ (Demilitarization Zone) relays communication. The relay server has its own firewall called iptables [3]. User process can configure iptables allow or deny communications specifying ip-address and ports making a chain, a group of rules. To build a tunnel through iptabes, it is necessary to know ip-address of client PC. Client PC has no way to know its ip-address, if it is behind of NAT (Network Address Translation). Http server receives client's ip address and launches a cgi-bin script which builds a tunnel through the firewall for the client's ip-address [4]. Firewalls outside of the relay server are already set to allow communication to the specific port of the relay server.

Two http processes serves for shift leader side and for outside. At the standby mode, one http server waits for shift leader's command. When shift leader make connection, another http server starts for outside.

Two http servers share data using a relational database management system, SQLite[5].

Zebedee[6] makes ip-tunnel with compression, encryption and authentication. It relays TCP and UDP protocol between two systems. Its name comes from zlib compression, browfish encryption and Diffie-Hellman key agreement. Those techniques are patent-free and distributed under GNU public license. Zebedee has been ported to many platforms we required.

VPN (Virtual Private Network) and SSH (Secure Shell) are popular tool for secure connection to the outside of the firewall. VPN enables users access control network as if they stay inside of the network. Shift leader hardly control their activity once they connect the network. This situation does not meet out requirement. SSH, as its name indicates, comes from interactive shell. It requires individual user's account on the server. We avoid SSH because many individual accounts may cause security problems. And a common account is more dangerous.

VNC shares screen and input devices of the workstation through the network. We did not use X11 to share control GUI, because it does not share screen and needs wide bandwidth to communicate. VNC runs on various platform and operating systems. VNC has variety of implementation. For example TightVNC [7] has efficient compression mechanism using JPEG, while original version [8] has no graphical compression mechanism. JPEG compression is fairly efficient at slow network environment. We choose an implementation called x11vnc [9] to share the screen of the control workstation. This implementation shares real X11 screen (:0.0 screen) while other VNC server create and share another X11 screen. We do not need re-start applications running on the workstations to share screen.

VNC-reflector [10] relays vnc protocol. It serves multiple client connection reducing VNC server's load. Though it might be rare case, multiple experts can access to one workstation through VNC-reflector.

Scripts written in Python [11] glues software components above together both in client and server side. Py2exe [12] program builds stand alone Windows programs from python scripts. Windows user needs only a client program made by py2exe instead of installing entire Python interpreter.

HOW WARCS WORKS

In the initial state, a http server is running at the relay server. And iptable firewall denies any connection from outside of the site. (Fig.2)

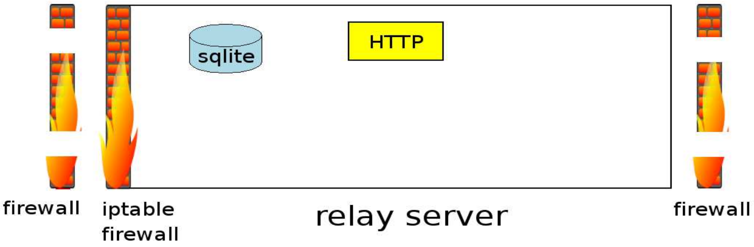

Fig2. Initial state of WARCS.

The shift leader calls an expert when the operation crew give up fixing the problem. The expert must have a mobile phone at the outside.

The shift leader accesses to the web page. The shift leader chooses communication application, VNC or telnet, choose target workstation to login and obtain generated one time password. Multiple client users, up to 10, can connect to the server. The connection is named for management. Iptable uses this name as the name of the chain. The shift leader launches vnc sever and zebedee server at the workstation. If graphical mode is chosen, zebedee allows vnc to connect only to the relay server, because the vnc server at the workstation is running in password-less mode for easy login. Nobody else but the relay server access the vnc server running at the workstation.

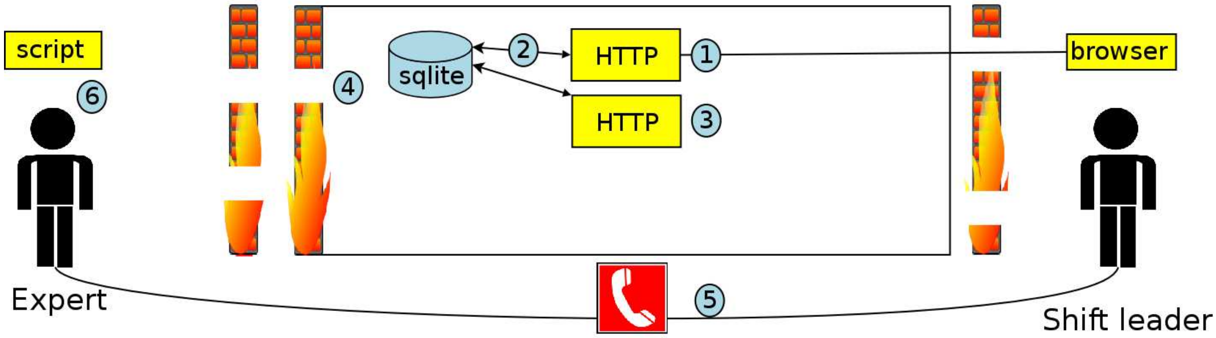

Fig.3 Shift leader begin to use WARCS.

The cgi-bin process started by web server do three jobs (Fig.3). One is generating eight-digit one time password (Fig.3-1) and displaying on shift leader’s web browser. The password is stored to a SQLite table in encrypted form (Fig3-2). Second is launching another web process which serves requests from the expert (Fig3-3). Third one is creating a iptable chain opening ip port for the web process (Fig3-4).

The shift leader tells the one time password to the expert by phone (Fig3-5).

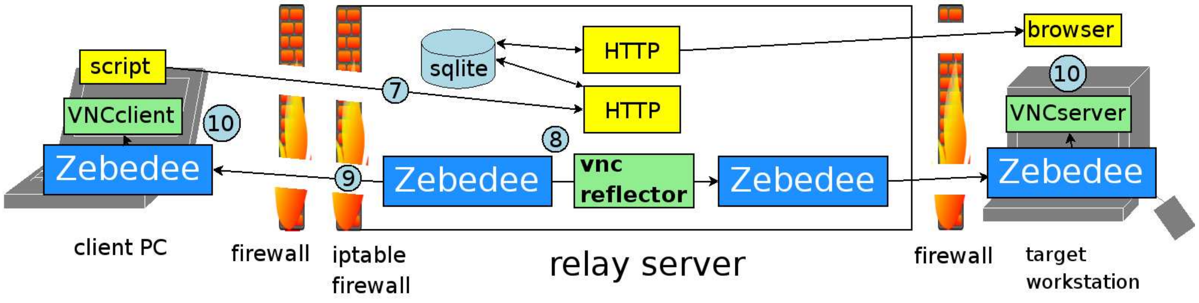

Fig4. Establish connection.

The expert connects a notebook PC to the internet and starts a client script written in python (Fig3-6). The script accepts an one time password from the experts and generates a secret and a public key for zebedee. The script sends the one time password and the public key to the web sever by http protocol (Fig4-7). If the password matches to the one store in the database the cgi-bin process start up zebedee process with public key authentication (Fig4-8). Those http communications are not encrypted. The cgi-bin process also adds ip-address and zebedee port to the iptable chain to allow communication through relay server's firewall (Fig4-9). In short, it digs a tunnel through the firewall. The zebedee server for outside client only accepts communication from the computer which has private key which is matched to the public key.

The client script starts up vnc client to display and zebedee process to communicate to the relay server (Fig4-10). The client script automates those complex processes. Just opening client process and entering one time password, the screen of the workstations appears on the PC's screen.

The shift leader can monitor status of connection accessing the database server via WWW during the experts is working. The shift leader can break the connection at any time from the web browser.

When the expert fixes the problem, he/she call the shift leader. The shift leader access a web page and click the end button on the page. A cgi-bin program removes corresponding chain from iptables and kills zebedee process. The shift leader also runs script to terminate vnc server and zebedee process. The relay server backs to the initial state.

OPERATION

We developed WARCS in 2003. Since then, every machine experts got training of operation before their trip. They have to test connection when they arrived. We also installed another relay server for test and training. The tests from outside has been done from various place including Europe and various internet providers. The machine experts also often use from their home at nigh time.

REFERENCES

[1] R. Tanaka et al., “Control system of Spring-8 storage ring“, ICALEPCS'95, Chicago, USA, 1995.
[2] A. Yamashita et al., “The database system for the SPring-8 storage ring control”, ICALEPCS’97, Beijing, China, 1997.
[3] Netfilter/iptable, http://www.netfilter.org/
[4] Ipshutter, http://www.wildspark.com/asher/ipshutter/
[5] SQLite, http://www.sqlite.org/
[6] Zebedee http://www.winton.org.uk/zebedee/
[7] TightVNC, http://www.tightvnc.com
[8] RealVNC,http://www.realvnc.com
[9] X11vnc, http://www.karlrunge.com/x11vnc/
[10] VNCReflector, http://sourceforge.net/projects/vnc-reflector/

[11] Python, http://www.python.org/
[12] py2exe, http://starship.python.net/crew/theller/py2exe/