BaroPAM Guide(Tomcat Console)

Index

1. BaroPAM integration

1.1 Introduction

Web Application Server (WAS) is a software framework that provides functions to create and operate web applications and server environments. It can be seen as middleware (software engine) that executes applications on the user's computer or device through HTTP on the Internet. A web application server is distinguished from a general web server by performing dynamic server contents, and is mainly performed like a database server. In Korea, it is commonly referred to as "WAS" or "WAS S/W", and is used as a "web application server" in public institutions, and is called "Application Server" (abbreviation AS) in English-speaking countries.
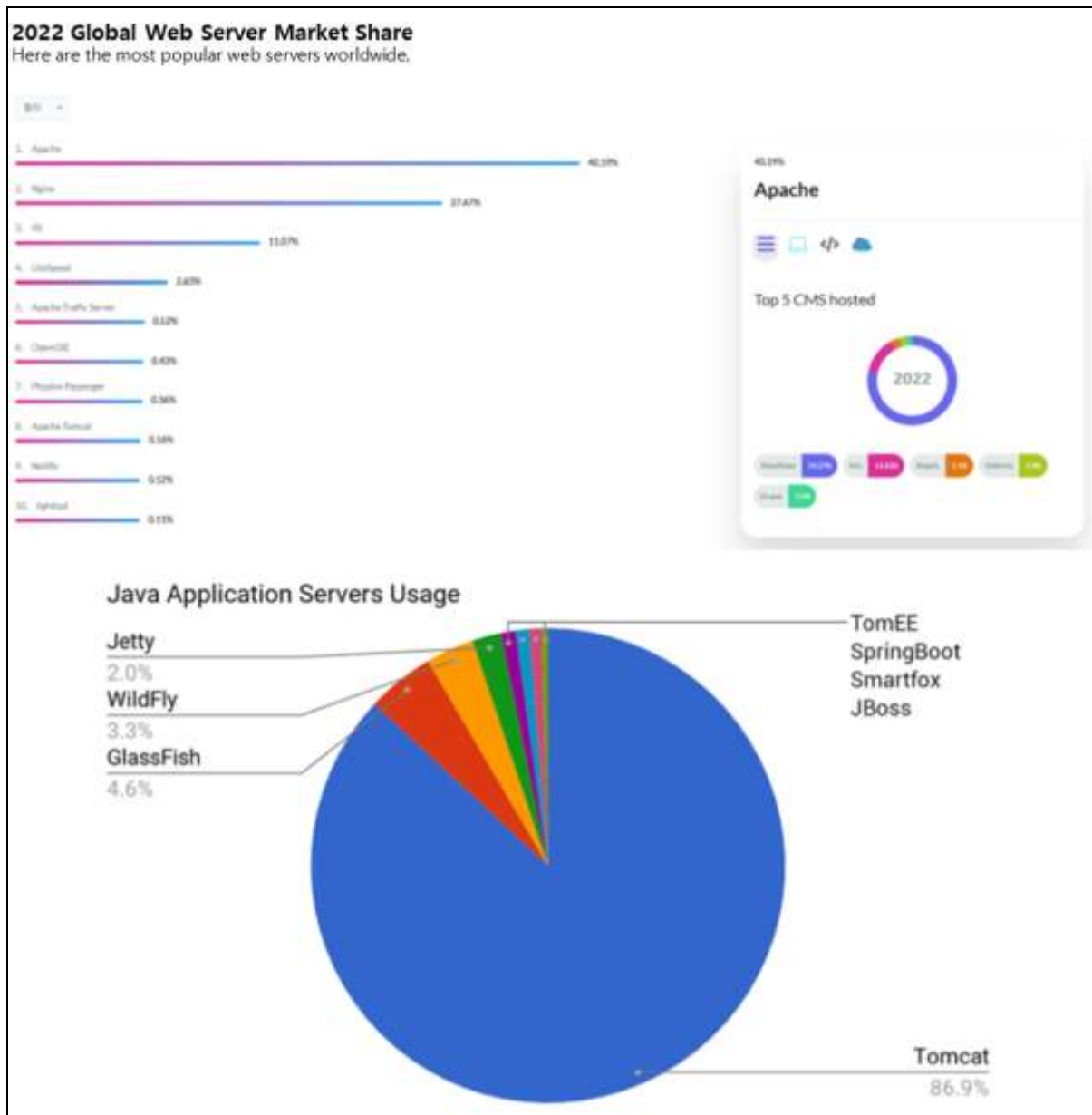
Most web application servers are based on Java and mainly accept Java EE standards, but there are also Java-based products that do not follow Java EE standards and non-Java products based on .NET or Citrix.

There are three basic functions of a web application server.

-Provides program execution environment and database access function.
-Manage multiple transactions.
-Perform business logic that handles tasks.

However, since there is no precise definition of web application, there are web application servers that do not provide some functions. In addition to these three features, vendors are adding and enhancing several features.

In order to create high-efficiency performance at low cost among web application servers, Apache Tomcat, an open source software, is being introduced in many places.

2022 Global Web Server Market Share
Here are the most popular web servers worldwide.

Java Application Servers Usage

In the past, Apache Tomcat was recognized as a field of limited software developers, but now it is secure enough to be widely used, and it has high versatility and outstanding performance as a leader among open source web application servers.

Apache Tomcat, an open source web application server, is an open source software of the open source foundation Apache Software Foundation, and is often used in conjunction with Apache, but sometimes it is used independently as a web service.

Hackers mainly use well-known attack techniques such as ▲ attacking application vulnerabilities ▲ malicious mail ▲ 'Watering Hole', which implants malicious code in a specific page on the website, to infiltrate the victim's PC in the early stages.

Among them, the most used is 'application vulnerability' attack. It is a method of attacking flaws in the security policy of application software such as MS Exchange, Oracle WebLogic, and Apache, or invisible vulnerabilities in system development.

Currently, it is widely used, so security is as important as it is, but design flaws such as

Tomcat's bugs and security vulnerabilities are used. It can also include an attack using a procedure or series of commands, scripts, programs, or specific pieces of data designed to perform an attacker's intended action.

The cause of this vulnerability attack affects the default settings of Tomcat, so it is highly likely that many servers are currently exposed to vulnerabilities. This is because users do not change the default settings very often. This means that many servers may be currently exposed to the Internet.

Recently, attacks targeting Tomcat's vulnerabilities are increasing. An attacker could exploit the vulnerability to run malicious code on a vulnerable server and gain control.

If these vulnerabilities are left unattended and operated, there are increasing cases of exploiting various vulnerabilities, such as attacking the vulnerabilities and executing malicious codes, distributing malicious codes, damaging operating servers, or looting data.

Ex) Administrator Console Vulnerability

```
-Tomcat provides the administrator console of the web environment.
-If the administrator console is invaded from the outside, all privileges related to the web may
 be leaked, so care must be taken in management.
-Vulnerable when the page for administrator authentication is easily recognizable or accessed by
 analogy.
-Security is vulnerable because Username/Password used for administrator authentication is stored
 in unencrypted plain text (tomcat-users.xml).
```

Ex) tomcat-users.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">

  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-gui"/>
  <user username="tomcat" password="tomcat" roles="tomcat,manager-gui"/>

</tomcat-users>
```

The Username/Password used for logging in to the Tomcat administrator console is stored in plain text, unencrypted, in the "**tomcat-users.xml**" file, making it vulnerable to security.

1.2 Install the BaroPAM module

Since the OTA(One-Time Authentication) key, which is the authentication code used by BaroPAM, is written based on Java, it can be applied to the latest **JDK 6.x or later**.

1) OTA Key Verification Module(barokey.jar)

The API that verifies the OTA key entered in the password field when logging in to the Tomcat administrator console is provided as "**barokey.jar**". You can place "**barokey.jar**" in the lib directory of WAS (Web application server) or set it by including the directory where "**barokey.jar**" exists in the classpath.
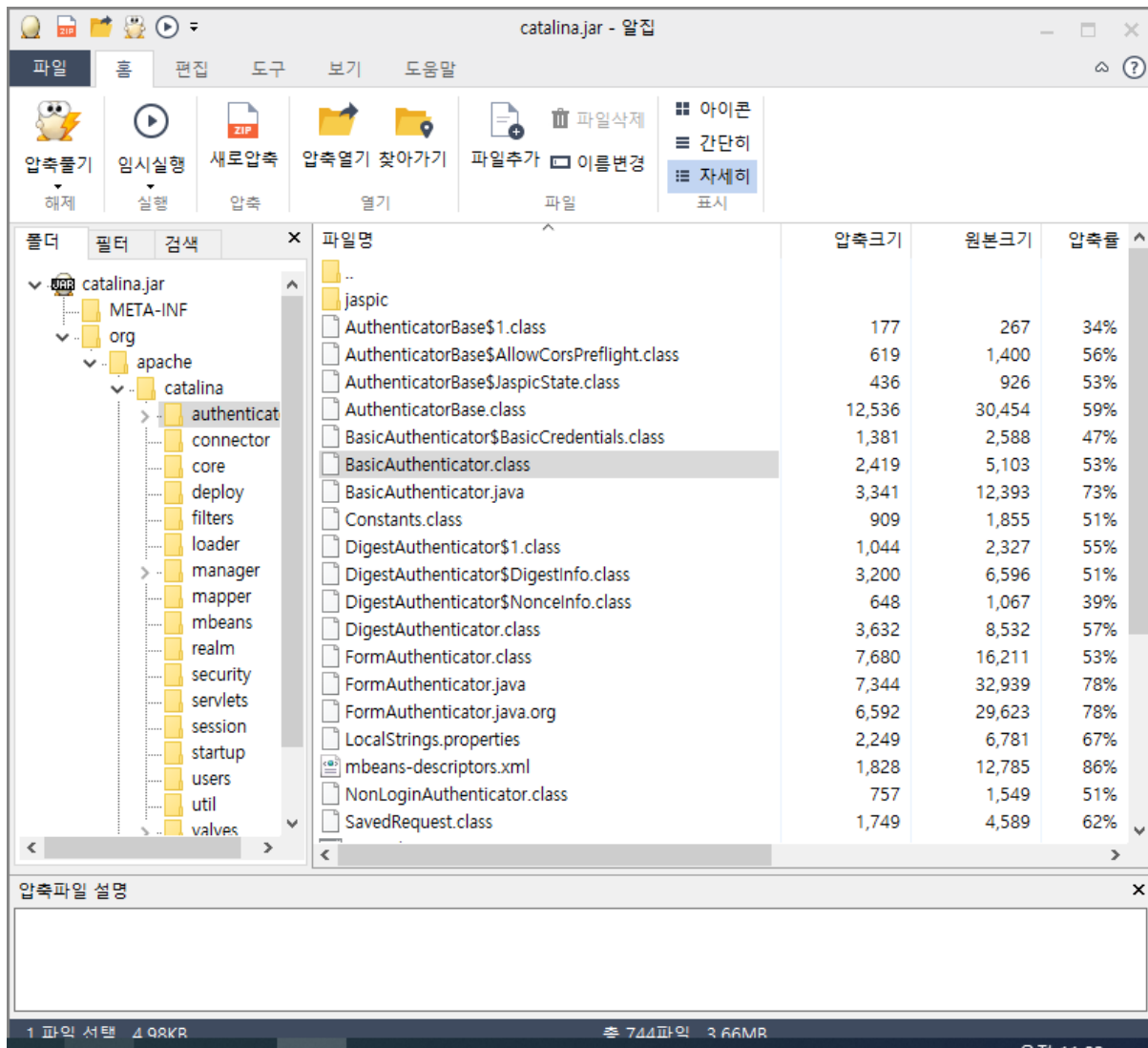
```
[root] /home/tomcat/lib > ls -al
total 23100
drwxrwxrwx  2 root root    4096 May 22 13:29 .
drwxr-xr-x 13 root root    4096 Aug 14  2017 ..
-rw-r--r--  1 root root    2550 Apr 23  2022 .bash_history
-rwxrwxrwx  1 root root   15240 Sep 18  2014 annotations-api.jar
-rw-r--r--  1 root root   16864 May 22 15:10 barokey.jar
-rw-r--r--  1 root root   16730 Mar 14 14:08 barokey.jar.old
-rwxrwxrwx  1 root root   54565 Sep 18  2014 catalina-ant.jar
-rwxrwxrwx  1 root root  132132 Sep 18  2014 catalina-ha.jar
-rwxrwxrwx  1 root root  237521 Sep 18  2014 catalina-tribes.jar
-rwxrwxrwx  1 root root 1243752 Sep 18  2014 catalina.jar
............
```

2) Tomcat manager console login module(BasicAuthenticator.class)

The "**BasicAuthenticator.class**" file related to logging in to the Tomcat administrator console. The existing jar file is "**catalina.jar**".

```
 [root] /home/tomcat/lib > ls -al
total 23100
drwxrwxrwx  2 root root    4096 May 22 13:29 .
drwxr-xr-x 13 root root    4096 Aug 14  2017 ..
-rw-r--r--  1 root root    2550 Apr 23  2022 .bash_history
-rwxrwxrwx  1 root root   15240 Sep 18  2014 annotations-api.jar
-rw-r--r--  1 root root   16864 May 22 15:10 barokey.jar
-rw-r--r--  1 root root   16730 Mar 14 14:08 barokey.jar.old
-rwxrwxrwx  1 root root   54565 Sep 18  2014 catalina-ant.jar
-rwxrwxrwx  1 root root  132132 Sep 18  2014 catalina-ha.jar
-rwxrwxrwx  1 root root  237521 Sep 18  2014 catalina-tribes.jar
-rwxrwxrwx  1 root root 1243752 Sep 18  2014 catalina.jar
............
```

Replace the "**BasicAuthenticator.class**" file in the "**org\apache\catalina\authenticator**" directory in the "**catalina.jar**" file, which is a jar file related to logging into the Tomcat administrator console, with a file that matches the Tomcat version.

1.3 BaroPAM configuration

"**BAROPAM**" and "**BAROCONF**" environment variables used in "**barokey.jar**", an interlocking API, must be set.

```
[root] /home/tomcat/conf > ls -al
total 372
drwxrwxrwx  3 root root  4096 May 23 10:44 .
drwxr-xr-x 13 root root  4096 Aug 14  2017 ..
-r--r--r--  1 root root    89 May  8 2019 .baro_nurit
-rw-r--r--  1 root root  5133 Apr 23  2022 .bash_history
drwxr-xr-x  5 root root  4096 Sep 18  2014 Catalina
-rw-r--r--  1 root root    23 Apr 20 11:14 .baropam.conf
-rwxrwxrwx  1 root root 10858 Sep 18  2014 catalina.policy
-rwxrwxrwx  1 root root  3794 Sep 18  2014 catalina.properties
-rwxrwxrwx  1 root root  2998 Jul 29  2022 context.xml
```

설정 예)

```
export BAROPAM=/home/tomcat/conf/.baro_nurit
export BAROCONF=/home/tomcat/conf/baropam.conf
```

Note) When changing the contents of the ".baro_nurit" file, a OTA key verification error occurs. In addition, ".baropam.conf" has a structure of "Secure key, Cycle time, Last login time, Logging flag(Y/N)".

1.4 Install BaroPAM app and set information

2. BaroPAM application

2.1 BaroPAM application process
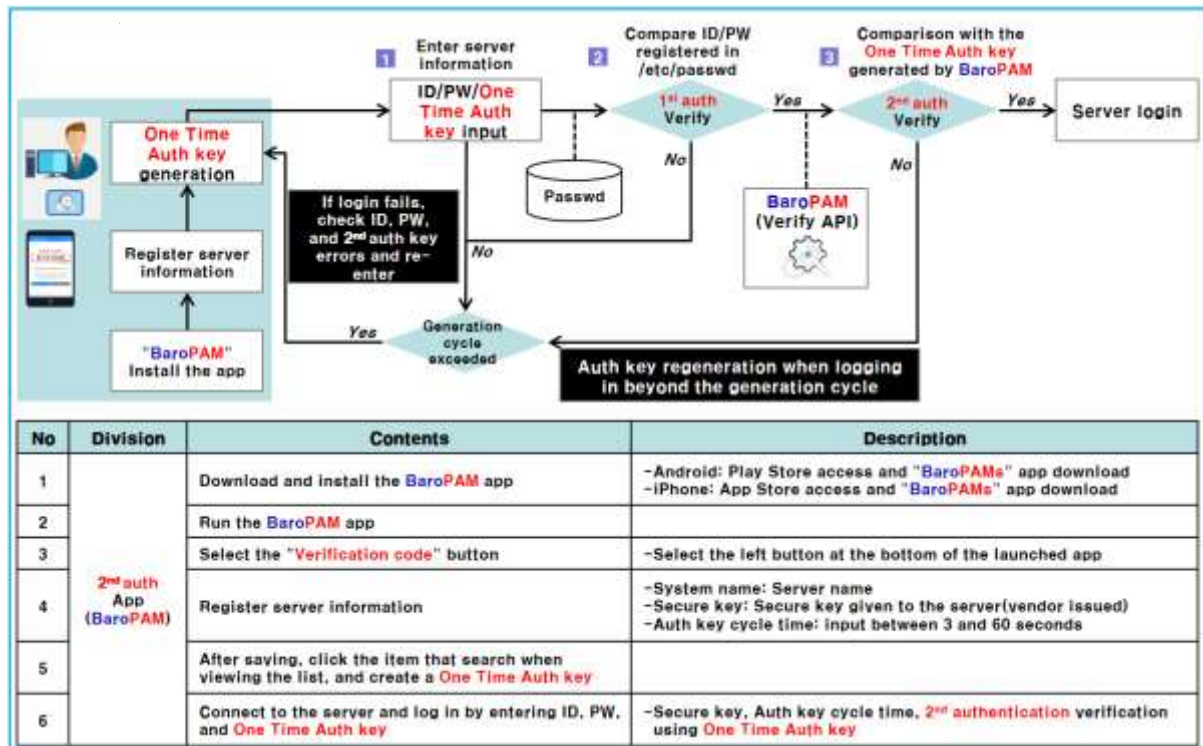
2.2 BaroPAM application screen

2.3 Login to Tomcat manager console

When entering passwords for all network devices such as Internet routers, wireless APs, IoT, routers, and switch equipment, you can apply the method of entering the password first and then entering the OTA key without spaces.

For example, if the password is "baropam" and the OTA key is "123456", enter "baropam123456" in the password field.

If the OTA key generation cycle is 30 seconds, the OTA key changes every 30 seconds, so when combined with a fixed password, a new password is generated every OTA key generation cycle. It eliminates the need to periodically change the password for the Tomcat administrator console, the classic method.

If the 2-factor authentication solution is applied, you cannot log in to the Tomcat administrator console if only a fixed password is entered in the password field.

Therefore, password alone is never safe when logging in to information assets, and a new application method (additional authentication, password replacement, new password) that can replace the password every time it is used is needed.

Until now, it seems that a change of mind is needed rather than changing a complicated password (appealing password syndrome/reset syndrome) according to password change regulations periodically once every quarter for security reasons.

Ex) Tomcat manager console
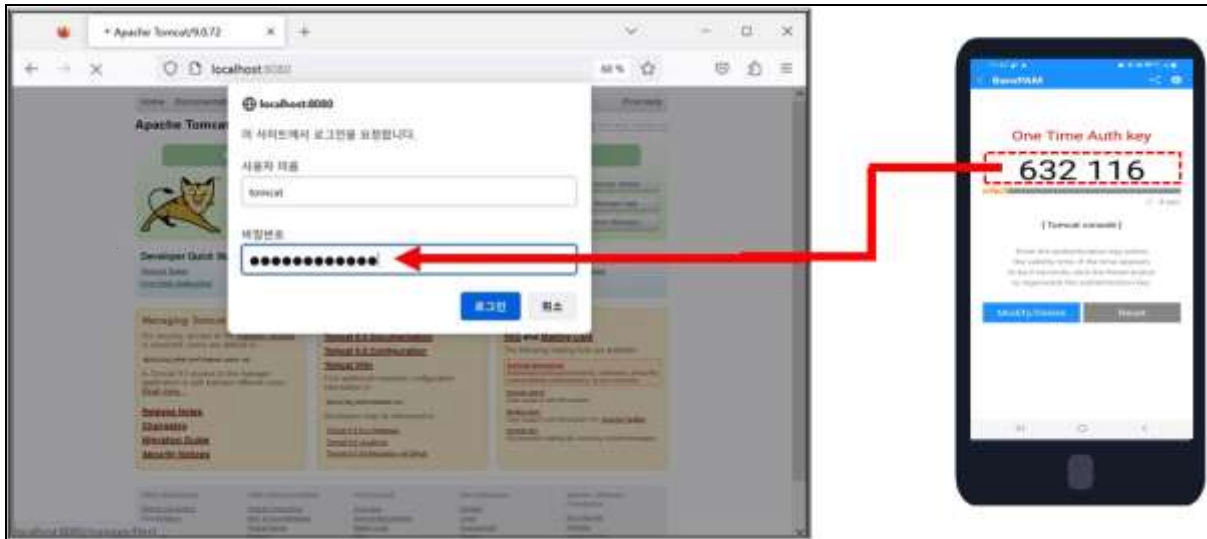
On the Tomcat console login screen, you must first enter your password, followed by a OTA key without a space. For example, if the password is "tomcat" and the OTA key is "632116", enter "tomcat632116".

Ex) Oracle Weblogic Server manager console
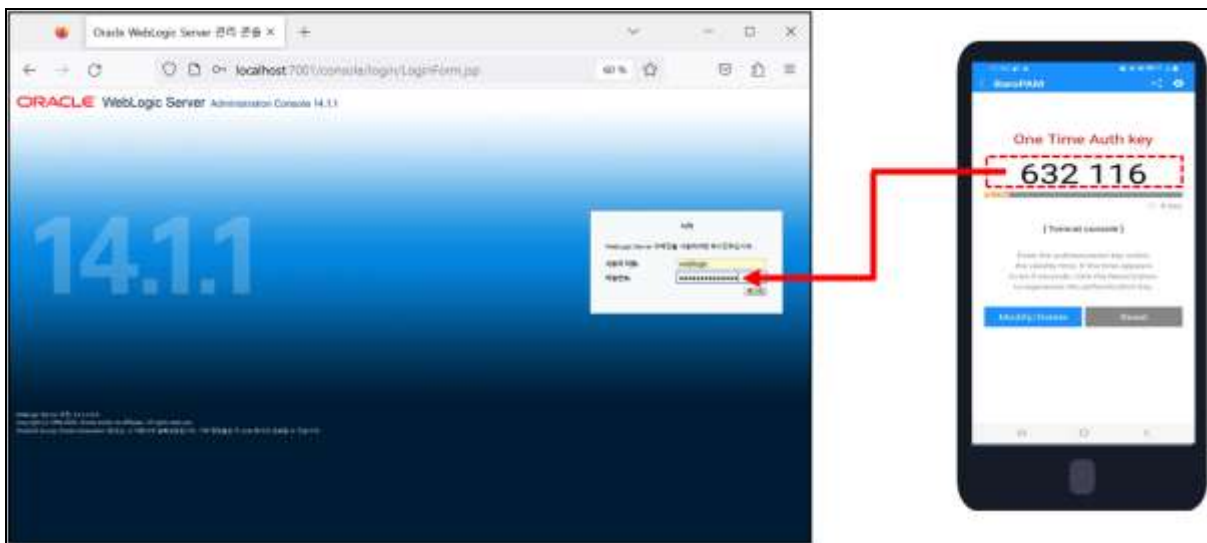
Even if Username/Password, which is the Tomcat console login information, is leaked, strong security can be applied because the OTA key added behind the password cannot be inferred.

3. About BaroPAM

Version 1.0 - Official Release - 2016.12.1
Copyright ⓒ Nurit corp. All rights reserved.
http://www.nurit.co.kr

Company: Nurit Co., Ltd.
Registration Number: 258-87-00901
CEO: Jongil Lee
Tel: +8210-2771-4076(Technical support, sales inquiry)
email: mc529@nurit.co.kr
Address: #913, 15, Magokjungang 2-ro, Gangseo-gu, Seoul (Magok-dong, Magok Techno Tower 2)