Behavior-based countermeasure against SSH Brute Force Attack

2008.11.13

TOMOYO Linux Project
Handa Tetsuo

TOMOYO is a registered trademark of NTT DATA CORPORATION in Japan.
Linux is a trademark of Linus Torvalds.
Other names and trademarks are the property of their respective owners.
• The SSH service, which is used for remote administration, can allow the attacker to leak secrets and/or trojan the system if the attacker successfully logged into the system.
  – Recently, the attacks are becoming more and more complicated and sophisticated, and sometimes defeats traditional defense approaches.

• This presentation demonstrates a brand new defense approach using "Operating Systems with Advanced Access Control Features".
  – I use TOMOYO Linux as an example of such OS.
• You know what this diagram is, don’t you?
• You know what this diagram is, don’t you?
Preparative: Example of STD in Linux

- Spreads like a tree from `/sbin/init`.
Preparative: What is TOMOYO Linux?

• A tool for designing and enforcing STD.
  – Monitors and judges program execution requests issued by userland applications.
  – Performs state transition by execution of a program.

• A tool for observing and restricting requests within each state.
  – Monitors and judges file’s read/write requests issued by userland applications.
  – Updates process’s internal state by read/write/execute requests issued by userland applications.
Preparative: Type of SSH sessions

• Interactive shell session
  – A shell is provided and the user can enter commands freely.
    • The user can access resources freely.

• Non-interactive shell session
  – Only commands passed to a shell’s "-c" option are executed.
    • scp and/or sftp

• Non shell session
  – A shell is not provided.
    • TCP port forwarding
Conventional approaches

- Authentication based on "What the user knows".
  - Thus, subjected to brute-force attacks.
- Assumes that login authentication is not run through.
  - Reduce possibility of being run through.
    - Banning failed clients for some period using firewalls.
    - Using public-key authentication.
  - Brute force attacks are getting distributed and secret-key are stolen by malwares.
Proposed approach

• Authentication based on "How the user acts".
  – Utilize state transition.
• Assumes that login authentication is run through.
  – Restricts after conventional login authentication.
  – Just gives the user a login shell and observes whether the user acts as expected or not.
  • Something like probation in employment contract.
Flow of conventional approaches

- Customizes until the login authentication.

1. SSH server accepts a connection from a client.
2. Do login authentication
   - Success: Do the user’s job.
   - Failure: SSH server disconnects the connection.
Flow of proposed approach

• Customizes after the login authentication.

SSH server accepts a connection from a client.

Do login authentication

Success

Observe the user’s behavior

Normal

Suspicious

Do the user’s job.

Failure

SSH server disconnects the connection.
Open the door!

• Don't be a slave to convention.
  – Nothing is taboo.
  – Implement your own ideas.

• The gate now opens.
  – Welcome to the secure world!
Case 1: Interactive shell session

- Utilize typing timing.
  - What we need
    - own program /bin/timeauth

```
/usr/sbin/sshd
/bin/bash
/bin/bash
/bin/timeauth
/bin/timeauth
/bin/bash
/bin/bash
```
Case 1: Interactive shell session

Authenticate me!
Case 1: Interactive shell session

Authenticate me!

Enter password.
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Case 1: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Case 1: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 1: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!

Enter password.

Password is PacSec
Case 1: Interactive shell session

Authenticate me!
Enter password.

Password is ******

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Enter password.

Password is PacSec

NG. Go away!

2008.11.13
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is

2008.11.13
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is P
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is Pa
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is Pac
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is Pa c S
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is Pa c Se
Case 1: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is Pac Sec
Case 1: Interactive shell session

Authenticate me!

Enter password.

Password is ******

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!

Enter password.

Password is Pa c Se c

OK. Go ahead.
Case 1: Interactive shell session

• Advantages
  – No limitations for available factors.
    • No need to follow standards like RFC.
    • No need to disclose factors you use.
  – Unimaginable authentic method.
    • You can use methods an intruder never expects.
    • Even shorter password, brute force becomes meaningless by combining password and typing timing information.
Case 1: Interactive shell session

• Disadvantages
  – You need "Operating Systems with Advanced Access Control Features".
    • To restrict commands executed from login shell.
    • Namely MAC (Mandatory Access Control).
  – Difficult to use if Round Trip Time is large.
    • It might be convenient for defending the system against foreign incursions or aggression?
Case 2: Interactive shell session

• Utilize one-time password (OTP) and mail.
  – What we need
    • SMTP server
    • own program /bin/mailauth
Case 2: Interactive shell session

Authenticate me!
Case 2: Interactive shell session

Authenticate me!

Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
Case 2: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!

Enter password.

Password is ******

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!

Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.

Authenticate me!
Enter password.
Password is ****
Case 2: Interactive shell session

Authenticate me!
Enter password.

Password is ******
OK. Authenticated.

Let me access shell.
OK. Go ahead.

Authenticate me!
Enter password.

Password is ****
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is

2008.11.13
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ********
OK. Authenticated.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is
OK. Go ahead.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.

Password is ******

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Case 2: Interactive shell session

Authenticate me!
Enter password.

Password is ***
OK. Authenticated.

Let me access shell.
OK. Go ahead.

Authenticate me!
Enter password.

Password is

2008.11.13
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is

2008.11.13

74
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ******
OK. Go ahead.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Case 2: Interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Let me access shell.

OK. Go ahead.

Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is *****
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ****
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ******

2008.11.13
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ****

2008.11.13
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ****
Case 2: Interactive shell session

Authenticate me!
Enter password.
Password is ******
OK. Authenticated.
Let me access shell.
OK. Go ahead.
Authenticate me!
Enter password.
Password is ♀
NG. Go away!

2008.11.13
Case 2: Interactive shell session

• Advantages
  – The process which generated OTP also verifies the OTP.
    • No need for time/counter synchronization mechanism.
    • OTP expires when the process dies.
  – No problem if OTP is leaked to anybody but the intruder.
    • OTP is valid for nobody but the user who created the process.
Case 2: Interactive shell session

• Disadvantages
  – You need to receive mail.
    • You need to carry a mail receiver (e.g. cell-phone).
  – You need to send mail.
    • You need to provide SMTP service or equivalent (e.g. sendmail CGI program on WEB server).
Case 3: Non-interactive shell session

• Utilize environment variables.
  – What we need
    • SSH server’s AcceptEnv directive
    • SSH client’s SendEnv directive
    • own program /bin/env_check
    • TOMOYO Linux’s execute_handler directive
Case 3: Non-interactive shell session

SSH server accepts a connection from a client.

Do login authentication

SSH server sets up environment variables.

SSH server executes execute_handler.

Validate environment variables.

Executes login shell.

Do the requested commands.

SSH server disconnects the connection.
Case 3: Non-interactive shell session

Authenticate me!
Case 3: Non-interactive shell session

Authenticate me!
Enter password.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is *****
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is ******

OK. Authenticated.

Let me access shell.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is ******

OK. Authenticated.

Let me access shell.

NG. Go away!
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Use this environ.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Use this environ.

OK. Ready.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is *****

OK. Authenticated.

Use this environ.

OK. Ready.

Let me access shell.
Case 3: Non-interactive shell session

Authenticate me!

Enter password.

Password is ******

OK. Authenticated.

Use this environ.

OK. Ready.

Let me access shell.

OK. Go ahead.
Case 3: Non-interactive shell session

• Advantages
  – Transparent from client’s point of view.
    • No need to modify command line.
    • No special handling for standard input/output.
  – You can use environment variables as password.
    • No need to disclose environment variable names.
    • You can assign different permissions according to the content of environment variables.
  – You can use this method for interactive shell session too.
Case 3: Non-interactive shell session

• Disadvantages
  – Available for only TOMOYO Linux.
    • I think only TOMOYO Linux supports `execute_handler` mechanism.
  – Possibility that a SSH client does not support `SendEnv` directive.
Case 3: Non-interactive shell session

- Example: Switch via environment variable.

```
/usr/sbin/sshd
```

```
Entrance
```

```
/usr/libexec/openssh/sftp-server
```

```
Login shell
```

```
/bin/bash
```

```
Read-write mode
```

2008.11.13
Case 3: Non-interactive shell session

/bin/env_chk

Revalidation

/usr/sbin/sshd

/bin/bash

Dummy

/bin/bash

Read-write mode

/usr/libexec.openssh.sftp-server

Read-only mode

/bin/bash

/usr/libexec.openssh.sftp-server

/bin/bash

/bin/bash

Dummy
Case 4: Non-interactive shell session

- Introduce original layer.
  - What we need
    - "-S" option of scp and sftp commands.
    - A server side program for monitoring behavior.
    - A client side program for controlling behavior.
    - TOMOYO Linux’s execute_handler directive.
Case 4: Non-interactive shell session

- Application layer
  - Behavior controller (own program)
    - /usr/bin/ssh
  - Application layer
    - Behavior monitor (own program)
      - /usr/sbin/sshd
Case 4: Non-interactive shell session

Application layer
Behavior controller (own program)
/usr/bin/ssh

Application layer
Behavior monitor (own program)
/usr/sbin/sshd
Case 4: Non-interactive shell session

Application layer

Behavior controller (own program)

/usr/bin/ssh

Application layer

Behavior monitor (own program)

/usr/sbin/sshd
Case 4: Non-interactive shell session

<table>
<thead>
<tr>
<th>Application layer</th>
<th>Application layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior controller (own program)</td>
<td>Behavior monitor (own program)</td>
</tr>
<tr>
<td>/usr/bin/ssh</td>
<td>/usr/sbin/sshd</td>
</tr>
</tbody>
</table>
Case 4: Non-interactive shell session

Application layer

Behavior controller
(own program)

1
3
4
5

Application layer

Behavior monitor
(own program)

1
2

/usr/bin/ssh

/usr/sbin/sshd
Case 4: Non-interactive shell session

Application layer

Behavior controller (own program)

/usr/bin/ssh

Application layer

Behavior monitor (own program)

/usr/sbin/sshd
Case 4: Non-interactive shell session

Application layer

/usr/bin/ssh

Application layer

/usr/sbin/sshd
Case 4: Non-interactive shell session

• Advantages
  – You can use factors which cannot be used by default (e.g. standard input/output, command line parameters).
  – You can combine this method with environment variables.
    • You can give more permissions to only clients which support this method.
Case 4: Non-interactive shell session

• Disadvantages
  – You need to prepare a client side program.
Why not implement using PAM?

• Degree of freedom and difficulty.
  – No interference with other PAM modules, for all factors (e.g. standard input/output, command line parameters) are dedicated to the own program.

• Everybody can develop their own programs.
  – Not restricted by standards like RFC.
Why not implement using PAM?

• No need to modify client programs.
  – PAM can’t be used unless the client supports it.
  – All clients can support programs executed after PAM (i.e. login shell).
Why not implement using PAM?

• You can make it mandatory.
  – No worry of being omitted by PAM configurations and/or other PAM module’s results.
  – No worry of loopholes (e.g. buffer overflows, OS command injection from login shell) because all possible STD patterns are defined and enforced by MAC.
  – You can use external program’s assistance casually.
• You can use your original protocol because this approach is a local authentication.
  – Your idea shields yourself from attackers.
• There are infinite factors available.
  – Thus, brute force is impossible unless correct behavior (STD) is kept secret.
• You can implement low-cost and low-impact methods.
Case 5: Non shell session

• Customizing client program.
  – What we need
    • Original SSH client program (e.g. JSCH).
    • TOMOYO Linux’s task.state keyword.

Application layer

/usr/sbin/sshd

Request order monitor (TOMOYO Linux)
Case 5: Non shell session

• Advantages
  – You can switch permissions without executing a program.
    • You can use the order of TCP port forwarding requests as password.
  – You can use this method for interactive and non-interactive shell sessions too.
Case 5: Non shell session

• Disadvantages
  – Available factors are limited.
    • You can’t rely on external program’s assistance.
  – Available for only TOMOYO Linux.
    • This method modifies TOMOYO Linux’s process state variables (i.e. task.state) without modifying the SSH server program (i.e. /usr/sbin/sshld).
  – You need to develop client program.
Case 6: Deploying on-demand honey pot

- You can redirect the intruder to honey pot. Of course, you can forcibly logout.
A paper is available.

• Chained Enforceable Re-authentication Barrier Ensures Really Unbreakable Security
  – In short, CERBERUS, the gatekeeper.
  – The content may be obsolete because it was written 3 years ago.
  – But the concept will be useful and applicable even now.