

【不具合報告】Nablarch HIDDENストア脆弱性について

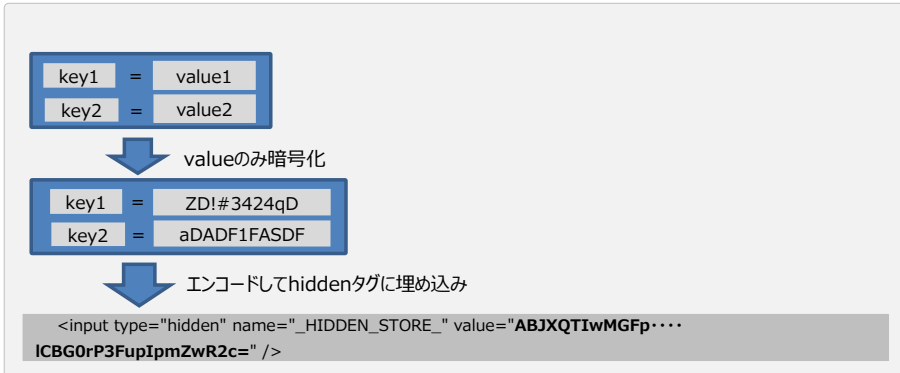
この度、NablarchのHIDDENストア機能について、後述する脆弱性が発見されました。
本報告書にて、脆弱性の内容、システム影響、および対応方法についてご報告致します。

1. HIDDENストア機能の概要

脆弱性が発見されたHIDDENストア機能は、HTMLのhiddenタグを使用して画面間でセッション変数を引き回す機能です。

key-valueオブジェクトのvalueのみ暗号化した後、エンコードした値をhiddenタグに埋め込みます。

また同様の機能（セッション変数を引き回す機能）としては、HTTPセッションストア、DBストアがあります。



Nablarch公開ドキュメント 「HIDDENストア」

https://nablarch.github.io/docs/5u13/doc/application_framework/application_framework/libraries/session_store.html#session-store-hidden-store

2. 本脆弱性について

2.1. 脆弱性の概要

HIDDENストア機能は、HTMLのhiddenタグに埋め込んだkey-valueオブジェクトの暗号化したvalueについて、
後述する特定の手段によってユーザが暗号化した値を予測・編集が可能になることから、改ざんの恐れがあります。

2.2. 本脆弱性が発生するNablarchのバージョン

Nablarch 5, 5u1～5u13を使用したWebアプリケーションが本脆弱性の対象です。

2.3. 本脆弱性の対象となるシステム

- ・ Nablarch5を使って実装したWebアプリケーション（インターネット・イントラネットは問わない）
- ・ 情報保持の実装方法にNablarchのHIDDENストアを使用している

2.4. 改ざんに必要な前提知識

攻撃者が改ざんを行うためには下記の情報を事前に把握している必要があります。

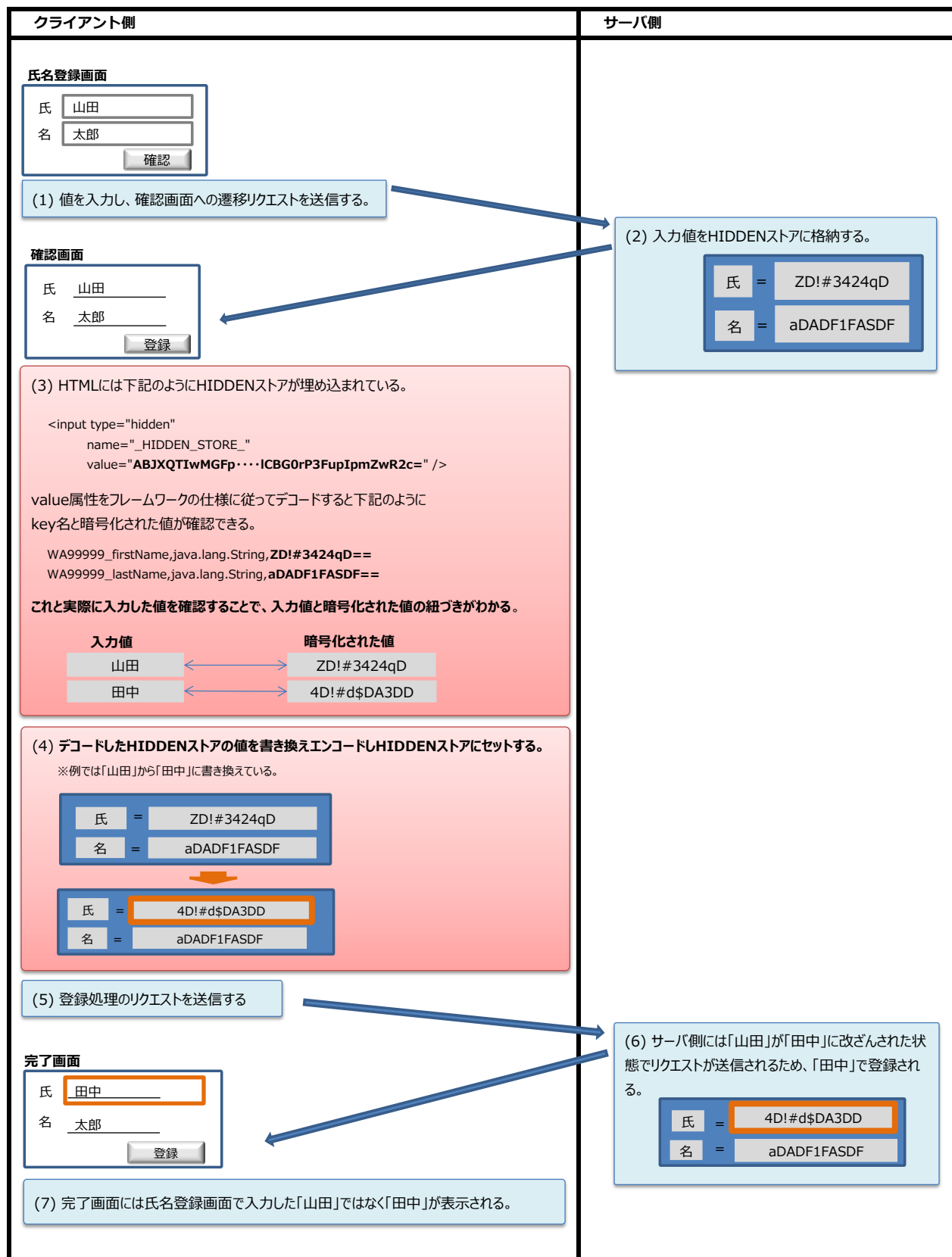
- ・ 対象システム(Webアプリケーション)がNablarchを使用していること
- ・ 対象システム(Webアプリケーション)において情報保持の実装方法にNablarchのHIDDENストアを使用していること
- ・ NablarchのHIDDENストアのデコード方法

2.5. 改ざん方法

HIDDENストアの改ざんは、HTMLのhiddenタグに埋め込まれる暗号化したvalueを予測・編集して実施します。

図中の赤色網掛部((3)、(4))が予測・編集を行っている箇所です。

なお図中の表記はあくまでもイメージです。HIDDENストアの値がイメージ通りに見えているわけではありません。



2.6. 本脆弱性によるシステム影響

本脆弱性への攻撃によるシステム影響は以下の通りです。

(1) 不正なデータ更新

HIDDENストアから取得した値をDB登録している場合、不正な値（バリデーションNGの値）が登録される恐れがあります。

※DB登録を行う前に、HIDDENストアから取得した値を再度バリデーションしている場合は本脆弱性の恐れはありません。

(2) 不正な画面操作

HIDDENストアから取得した値を画面遷移や画面表示切替に使用している場合、本来操作不可能な操作をされる恐れがあります。

※画面遷移情報の引き継ぎで使用している場合は、認可制御を適用していれば不正遷移は防ぐことが可能です。

【補足】HIDDENストアに格納するデータ構造による本脆弱性の発生有無・影響度

HIDDENストアに格納する値の性質により、改ざんの難易度が変わります。

・単一の値を保持するオブジェクト(例:文字列や数値)を格納する場合

改ざんの可能性が高いです。

別の箇所で入手した値を流用して改ざんされる可能性が高いです。

例えば、バリデーション条件が緩い入力項目から入手した値を

よりバリデーション条件が厳しい、別の入力項目に流用される恐れがあります。

値の型が汎用的であるほど（例:java.lang.Stringやjava.lang.Integer）、流用による改ざんが容易になります。

・複数の値を保持するオブジェクト(EntityやDTO等)の場合

改ざんの可能性が低いです。

オブジェクトが持つ値の一部を書き換えることはできないため、オブジェクト全体を置き換えることしかできません。

通常、EntityやForm等はバリデーション済みの値を保持しているため、

不正な状態を持つオブジェクトを入手すること自体が難しくなります。

また、改ざんした値が改ざん前の値と型が合わない場合は改ざん失敗となるため、

ある画面で取得した値を全く別の画面に流用できる可能性は低くなります。

3. 不具合原因と対応方法（Nablarch）

本脆弱性の発生原因は、下記2点です。

- ・HIDDENストアに格納するkey-valueオブジェクトのvalueを個別に暗号化している

これにより、暗号化後のvalueの値が容易に取得可能です。

対象ソース

リポジトリ	nablarch-fw-web
クラス名	nablarch.common.web.session.store.HiddenStore
リンク先	https://github.com/nablarch/nablarch-fw-web/blob/1.5.0/src/main/java/nablarch/common/web/session/store/HiddenStore.java

修正内容

- ・暗号化後の値をデコードして予測・編集できないように、value個別ではなくkey-valueオブジェクト全体で暗号化します。
- ・上記修正に加えて、より予測・編集できないようにするために、システム利用者ごとに異なる情報を用いて暗号化処理を実施しています。
- ・具体的なソースコードの修正内容は以下参照。

<https://github.com/nablarch/nablarch-fw-web/blob/1.5.1/src/main/java/nablarch/common/web/session/store/HiddenStore.java>

4. 対応方法（Nablarch利用プロジェクト）

4.1. 基本的な対応方法

本脆弱性の対象となるシステムは、最新版へのバージョンアップを行ってください。
バージョンアップにより、本脆弱性が解消できます。

4.2. バージョンアップが困難な場合の回避方法

2つの方法があります。

1. HIDDENストアから取得した値を精査するように改修

HIDDENストアの値の改ざんされるリスクを想定して、
HIDDENストアの値取得時に精査処理を必ず実装するように改修してください。

2. 5u14のHiddenStoreを参考に脆弱性対応済のHiddenStoreを作成し、それを使用するように改修

推奨はしませんが、5u14にて脆弱性対応済のHiddenStoreを参考にして、
プロジェクト側で、脆弱性対応済のHiddenStoreを作成し、それを使用するように改修してください。
ただし、脆弱性対応済のHiddenStoreはNablarch 5u14でしか動作検証を実施していないので、
作成したHiddenStoreがプロジェクトで使用しているNablarchのバージョンで正常動作するか必ず検証してください。