TECHNICAL PAPER

# Credential Stuffing Attacks

Last update: November 2021

# Contents

## Relevant Products and Releases

- SAS® Fraud Management 6.1

## Overview

Credential stuffing is a cyber-attack similar to a brute-force attack. Whereas, in a brute-force attack, the fraudster is essentially guessing a username and password by trying all possible combinations. In credential stuffing, the fraudster has gotten a list of usernames and associated passwords from a data breach and uses them to attempt to log on to an unconnected organization.

The enforcement of a strong password can successfully slow a brute-force attack so that it is no longer viable, but the additional exposed information that is used in a credential stuffing attack means that it has a higher success rate than the traditional brute-force attack. The fraudster is counting on the fact that users have reused their credentials across multiple sites. Unfortunately, this is often the case, despite ongoing attempts to educate the public about this threat. If the same password is used for multiple accounts, then that password strength provides no protection against a credential stuffing attack when one of those accounts has been compromised.

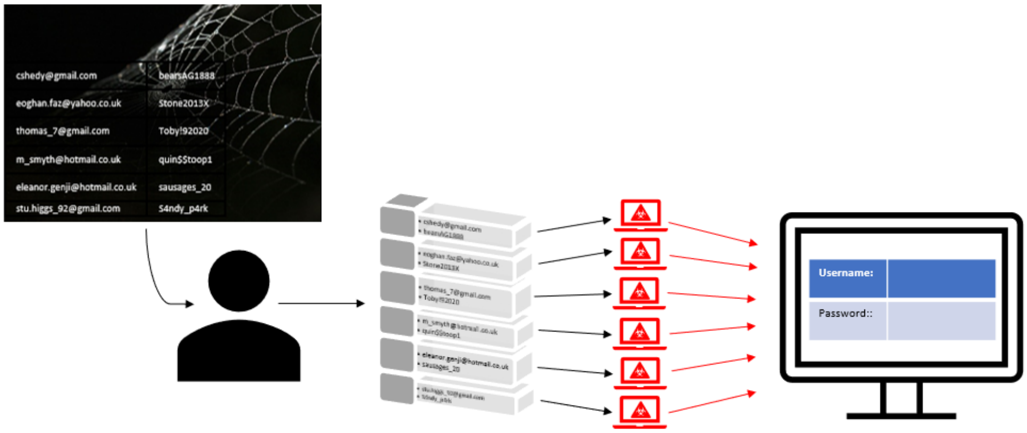
Generally, these attacks have low success rates, but with increasingly sophisticated bot technology, a fraudster is able to scale the attack with little effort. If you consider the millions, sometimes billions, of credentials that are used, even a small percentage of successful attempts can make this method of attack worthwhile for the fraudster.

## What Is the Modus Operandi?

The fraudster obtains a list of stolen usernames and associated passwords. They might steal these details themselves, or they could buy the credentials on the dark web where the number of these lists are growing as new data breaches continue to occur.

This list is then fed to a botnet that simultaneously attempts multiple logons to the online user accounts of a targeted organization.

*__Figure 1.__ Modus Operandi*



| | |
|---|---|
| cshedy@gmail.com | bearsAG1888 |
| eoghan.faz@yahoo.co.uk | Stone2013X |
| thomas_7@gmail.com | Toby!92020 |
| m_smyth@hotmail.co.uk | quin$Stoop1 |
| eleanor.genji@hotmail.co.uk | sausages_20 |
| stu.higgs_92@gmail.com | S4ndy_p4rk |

The fraudster monitors the results for successful logons and when this occurs, they know that they have a valid set of credentials.

## What Is Your Defense?

You can encourage your customers to create a unique password for your service. However, it isn't guaranteed that they will follow your guidance. You need to take your own measures to protect your organization.

Implementing additional security logon features (such as two-factor authentication or captchas) can help to prevent credential stuffing attacks, but they can also add friction to the customer experience.

However, if you combine these measures with analytics so that they are introduced only in certain suspicious scenarios (such as abnormal customer behavior or when there been a high number of logon attempts from an IP address or a device), then there is less impact to the average customer.

You can also add authentication steps into other parts of the process (not just at logon), then even if the fraudster has been able to gain access to an account, these extra layers of security can prevent money being removed from the account.

Having a system to quickly notify the customer of a logon attempt or suspicious behavior can also help to prevent the losses from a credential stuffing attack. Ideally, this should be a two-way system so that once notified, customers can indicate that the logon was not theirs and their account can be immediately disabled so that no time is wasted trying to contact your organization or to log on themselves and change their password.

# Using SAS Fraud Management to Detect Credential Stuffing Attacks

## Overview

A cyber security solution or bank system are expected to defend against logon attempts for invalid user accounts. However, for the instances where these attacks hit on valid accounts, SAS Fraud Management can provide an additional line of defense.

There are several ways that SAS Fraud Management rules could be used to detect credential stuffing attacks. For example, customer profiles could be developed to detect anomalies in the customer's logon behavior, lists of compromised email addresses could be collected from public data dumps and put in a lookup list to heighten the risk of logons on any of those potentially compromised accounts, and the flexibility of the rule response can be used to layer in the additional authentication steps, if needed.

In this paper, tracking a number of logon attempts from different user IDs that occur on a device in a short period of time is examined. The rule parameters (for example, the number of user IDs that are being tracked and the time period) and rule actions can be altered as needed.

## Create a User Variable Segment

The rule that you will create tracks information about logon attempts on a device. You will first need to create a user variable segment to uniquely identify the device. To ensure that the device is uniquely identified, select suitable key fields for the user variable segment. You can select multiple fields as long as they do not total more than 100 bytes.

If you do not choose a suitable user variable segment key, then the system might experience cardinality contention due to data volumes. The locking of the user variable segment key record, which is often required to ensure the accuracy of data, prevents other transactions from using the same record. Only one transaction can update the row at a time, so other transactions must wait until the current transaction is finished. Although this takes milliseconds, a build-up of waiting transactions can occur. Cardinality issues occur when multiple transactions are trying to update the same user variable segment key within a short period of time.

*Note*: The user variable segment prefix that is used in the following rule code is D. If you are already using the prefix D, then you can choose another prefix, but the rule code needs to be updated to replace _d_ with the user variable prefix that you chose.

Remember to test new user variable segments for the effects of cardinality, should it occur. Even without the impact of cardinality, the maturation of a user variable segment, once introduced, can have technical considerations that are not apparent to the bank. Consider the deployment of the user variable segment (even without a rule that references the new user variable segment).

Before implementation, consult with your internal IT department and SAS experts, including your Technical Support Account Manager (TSAM).

## Create User Variables

Now that you have created a user variable segment, you can create the user variables to use in the rule.

*Note*: You can use the replication count feature to create copies of the user variables.

***Table 1.*** *User Variables*

| Name | Type | Initial Value |
|---|---|---|
| _d_login_id_1 | Character 56 | Not applicable |
| _d_login_id_2 | Character 56 | Not applicable |
| _d_login_id_3 | Character 56 | Not applicable |
| _d_login_id_4 | Character 56 | Not applicable |
| _d_login_id_5 | Character 56 | Not applicable |
| _d_login_id_6 | Character 56 | Not applicable |
| _d_login_id_7 | Character 56 | Not applicable |
| _d_login_id_8 | Character 56 | Not applicable |
| _d_login_id_9 | Character 56 | Not applicable |
| _d_login_id_10 | Character 56 | Not applicable |
| _d_login_id_11 | Character 56 | Not applicable |
| _d_login_dt_1 | Datetime | 0 |
| _d_login_dt_2 | Datetime | 0 |
| _d_login_dt_3 | Datetime | 0 |
| _d_login_dt_4 | Datetime | 0 |
| _d_login_dt_5 | Datetime | 0 |
| _d_login_dt_6 | Datetime | 0 |

| Name | Type | Initial Value |
|---|---|---|
| _d_login_dt_7 | Datetime | 0 |
| _d_login_dt_8 | Datetime | 0 |
| _d_login_dt_9 | Datetime | 0 |
| _d_login_dt_10 | Datetime | 0 |
| _d_login_dt_11 | Datetime | 0 |

## Create the Variable Rule

The variable rule stores the datetime and user IDs for the last 11 new user ID logon attempts on a device.

Copy the following rule code into your variable rule:

```
/*Array of UV to store the last 11 different User IDs that
have attempted to log on using the device*/
%DECLAREARRAY(&rule.last_11_id_login ,_d_login_id_1 - _d_login_id_11);
/*Array of UV to store the date and time of the last 11 logins
on the device for different user IDs*/
%DECLAREARRAY(&rule.last_11_login_dt ,_d_login_dt_1 - _d_login_dt_11);

if TNG_TRAN_TYPE = 'OL' /*logon attempt*/
and %INDEXARRAY(&rule.last_11_id_login,hqo_ob_userid) < 1 /*new user ID logon*/
then do;
    %SHIFTHISTORYARRAY(&rule.last_11_login_dt);
    %SHIFTHISTORYARRAY(&rule.last_11_id_login);

    %SET(_d_login_dt_1) = dhms(rqo_tran_date,0,0,rqo_tran_time);
    %SET(_d_login_id_1) = hqo_ob_userid;
end;
```

## Create the Authorization Rule

The authorization rule checks whether there have been more than 10 logon attempts for different user IDs within the last 5 minutes. If the criteria are met, then additional verification is asked for and an alert is created.

Copy the following rule code into your authorization rule:

```
if tng_tran_type = 'OL' /*logon attempt*/
and dhms(rqo_tran_date,0,0,rqo_tran_time) - _d_login_dt_11 <= hms(0,5,0)
then do;
   %ACTION_VERIFY;
  %ACTION_ALERT;
end;
```

When selecting the alert type, choose a device-level alert type, if available. This alert view enables your fraud analyst to see all the events that occur on the device, which helps them identify whether it is a credential stuffing attack.

After you have created both rules, follow your normal procedures for testing and promoting the rules.

For more advice and support about preventing credential stuffing attacks and implementing strategies, contact your local or global fraud expert at SAS.

# References

SAS Fraud Management documentation is intended for use by existing customers and requires an access key. You can obtain the access key from your SAS consultant or by contacting SAS Technical Support. To expedite your request, please include SAS Fraud Management in the subject field of the form. Be sure to provide the SAS Site Number for your software license along with your request.

**Release Information** Content Version: 1.0 November 2021

**Trademarks and Patents** SAS Institute Inc. SAS Campus Drive, Cary, North Carolina 27513

To contact your local SAS office, please visit: sas.com/offices