Targeting RSA Emergency Access Tokencodes for Fun and Profit

written by Alexander Polce Leary | June 13, 2017

SecurID Emergency Access Tokencodes

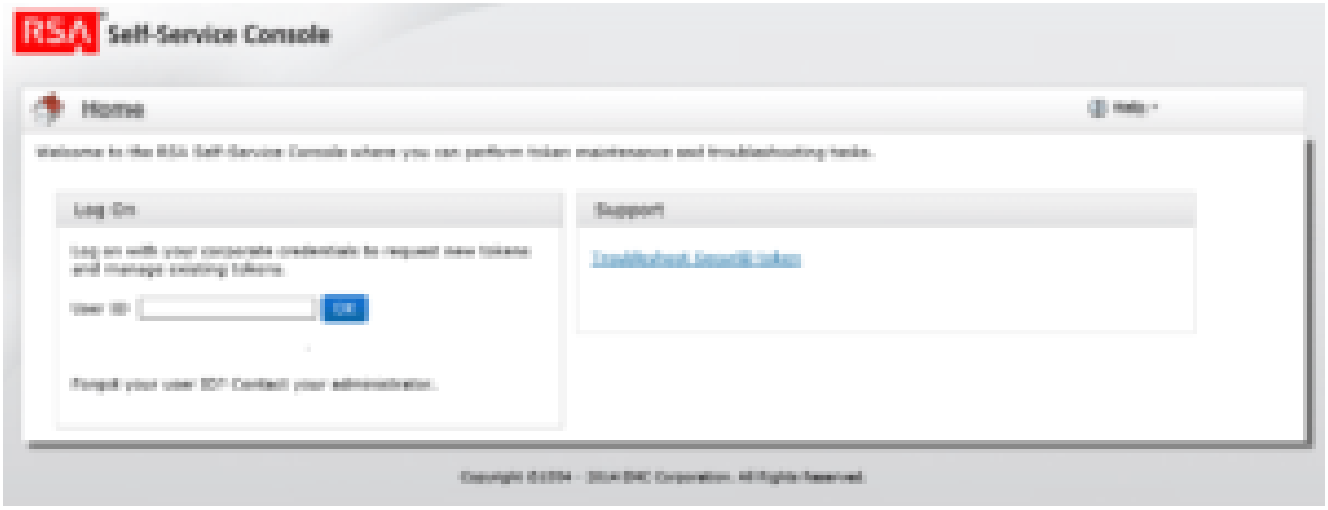
A few months ago, one of my RSA soft token was on the fritz. It refused to work, and I was not able to remote into the client's network to do an internal project for them. In fiddling with the RSA self-service console, and playing around with the troubleshooting section, I came across this feature called the Emergency Access Tokencode.

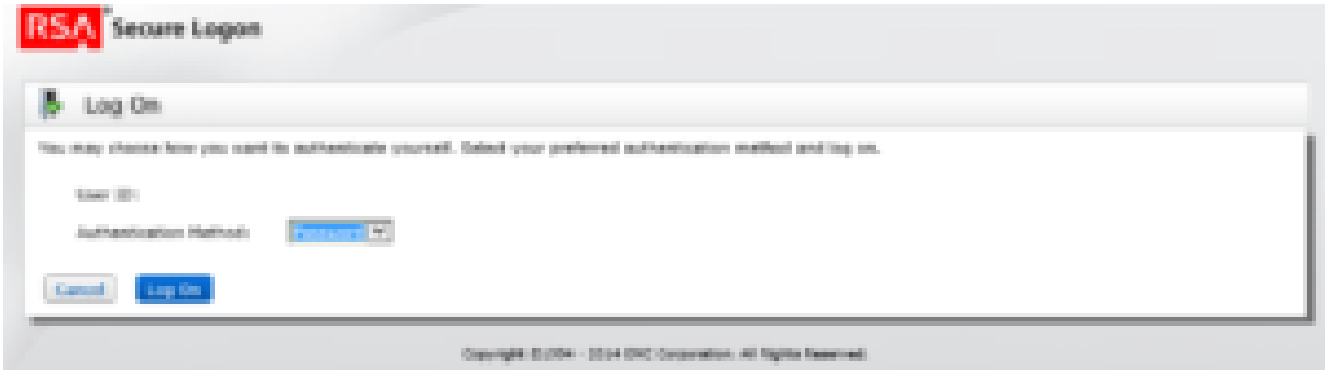
Hmmm I wonder what that is?

The Emergency Access Tokencode (EAT), is a backup code that is randomly generated on the RSA server that works for a set period, typically a week or so. Awesome, I didn't need my soft token anymore, and I'm on the client network legitimately.

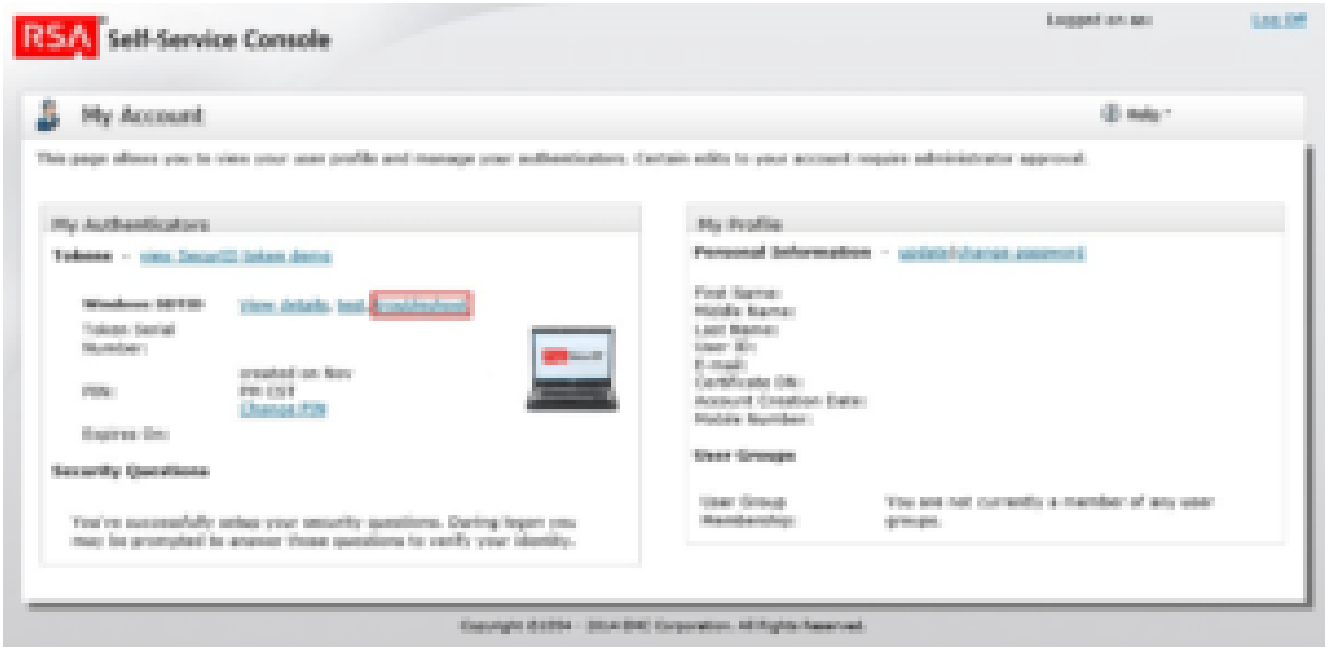
A few weeks later I was on an internal assessment for a PCI internal pen test. The CDE was isolated from the user and server networks via a Jump Host that used, you guessed it, RSA SecureID tokens for a second form of authentication. This is how I used the self-service console to get a legitimate token to bypass the 2FA.

RSA Self-Service Consoles have the option to integrate with LDAP for authentication. I have seen this commonly implemented, despite the potential pitfalls. After compromising a user's account who had access to the CDE, I was able to log into the RSA console using their Active Directory username and password.



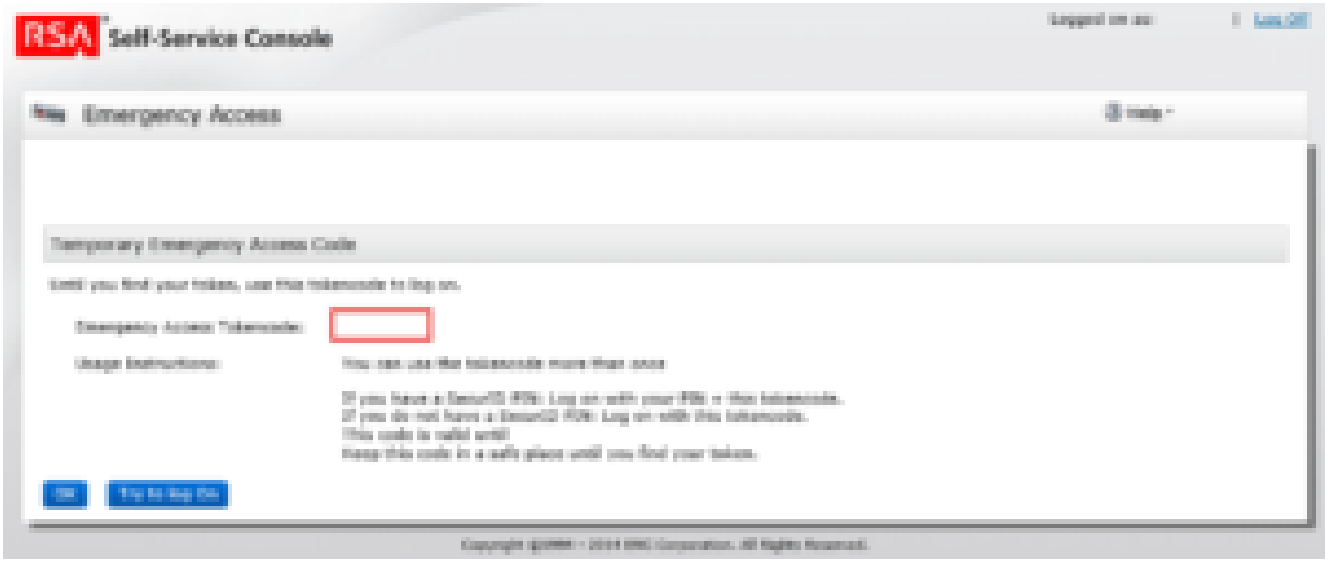


Once inside the console, navigate to the troubleshooting page.



And select the option, “Token is temporarily unavailable or misplaced”

At this point, an Emergency Access Tokencode was issued that was valid for a week.



In this instance, the user was not using a pin with the token, which allowed for direct access. However, this is not always the case though. In instances where there is a pin set, a work around is possible. Within the RSA console, there is a Change PIN field.

Normally a change pin field wouldn't be particularly interesting, however there are a couple abnormalities to it.

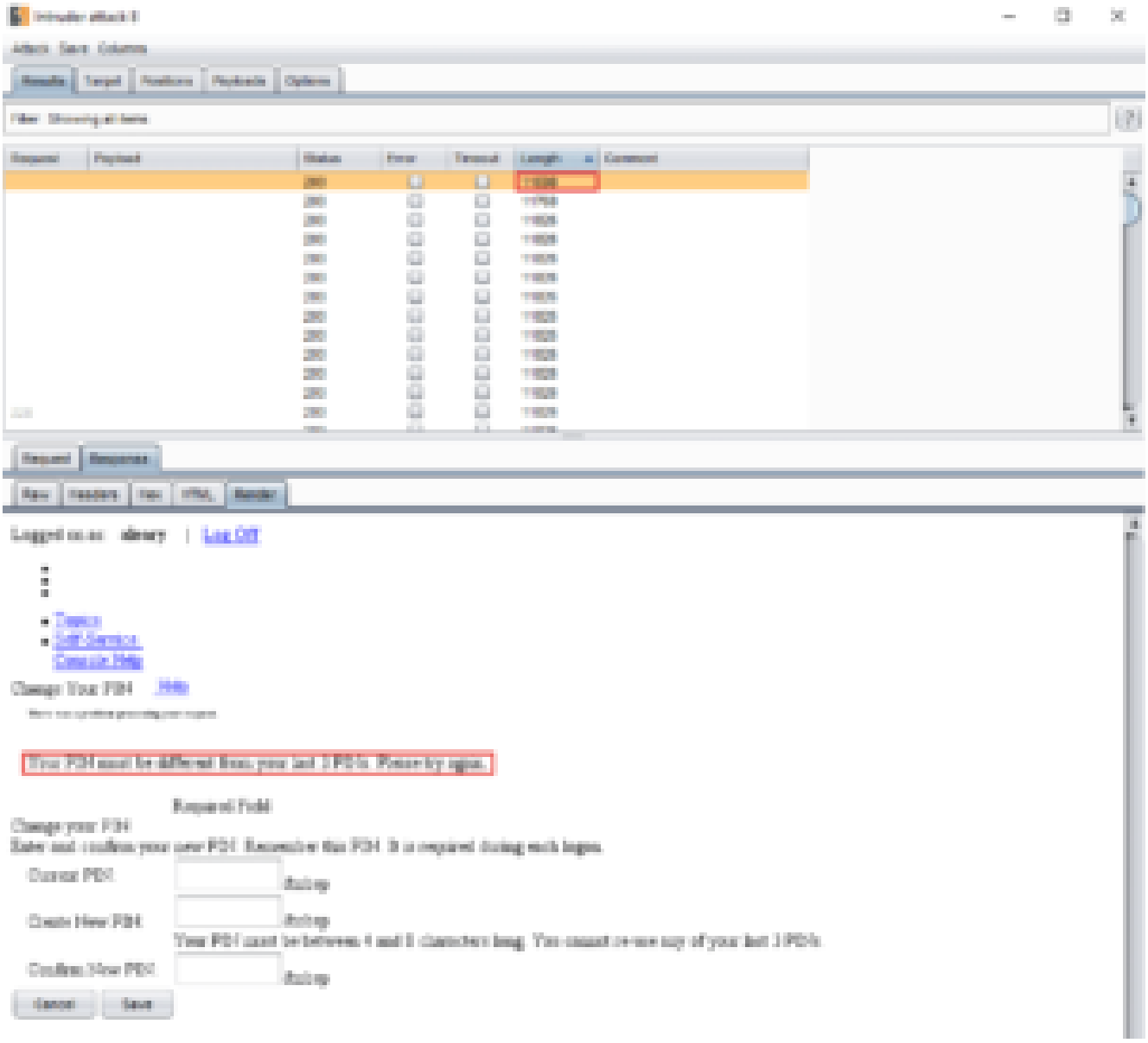
- The PIN is a finite length
- The PIN is all numbers
- The change PIN field is not governed by the standard lockout policy

What does this add up to?

This field is in most instances bruteforceable, and you're basically guaranteed to get the PIN. The longer the PIN, the more time it takes to brute force. However, if human nature holds true, most of the time users will make a pin that is the minimum length of 4 or 6 character, both of which can be brute forced in a few minutes.

Using Burp Suite's Intruder Attack, with the Battering ram attack type, we can discover what the current PIN is without changing it.

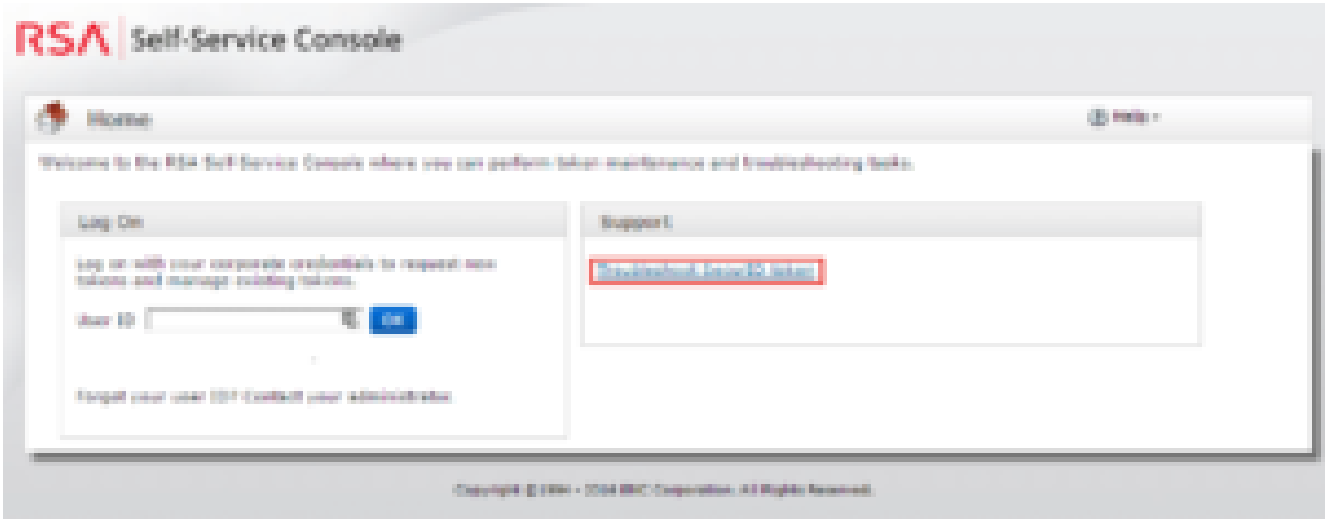
By sorting attack responses for length, we can find the anomaly, which is from a different error.



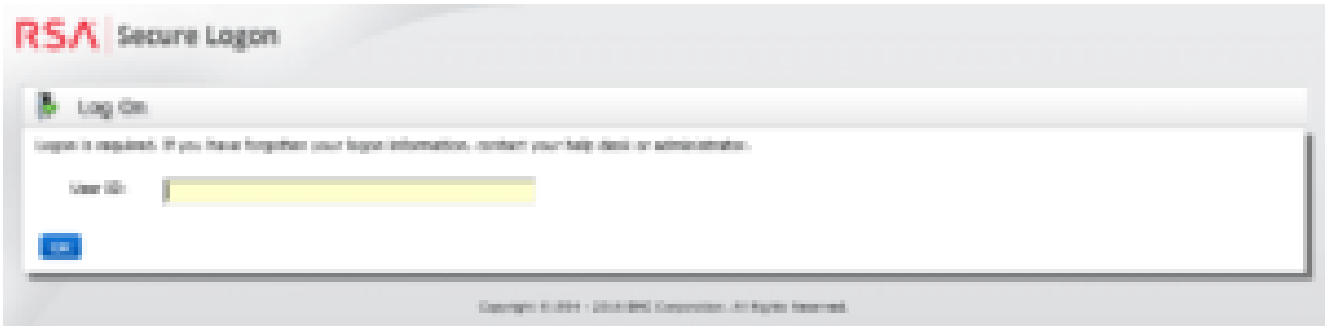
Now that we have our backup token and PIN we can now log into any SecureID protected system.

In instances where the Self-Service Console is not integrated into LDAP, and we also were not able to retrieved the saved password from a browser, we still have a one more card to play before we need to move onto social engineering.

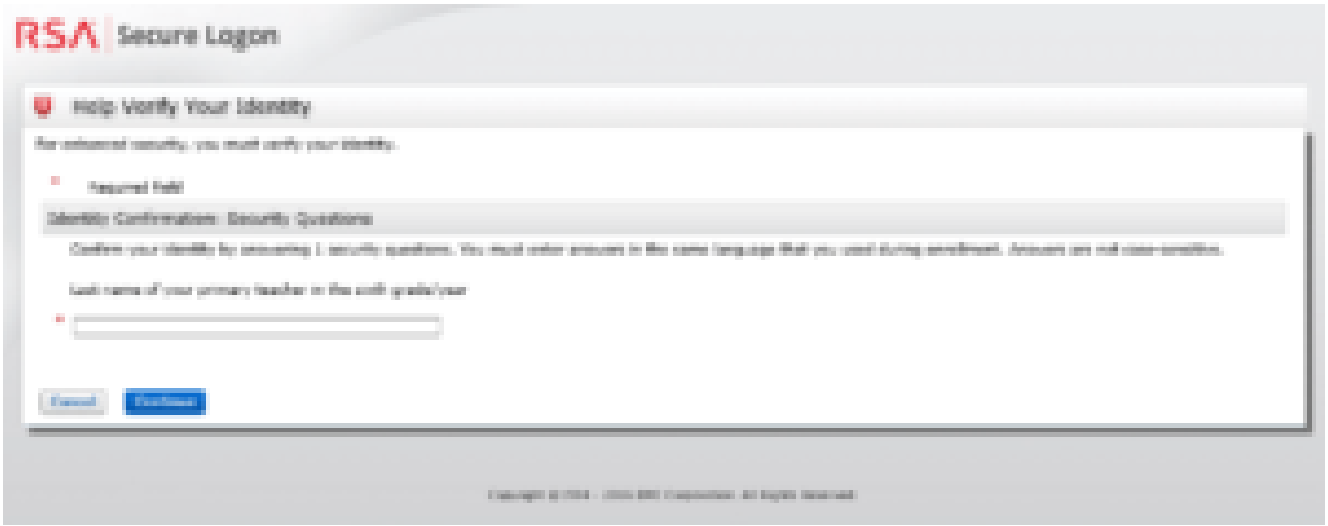
Going back to the login page we can see a “Troubleshoot SecureID token” link.



If we follow the link we find a input for a username.



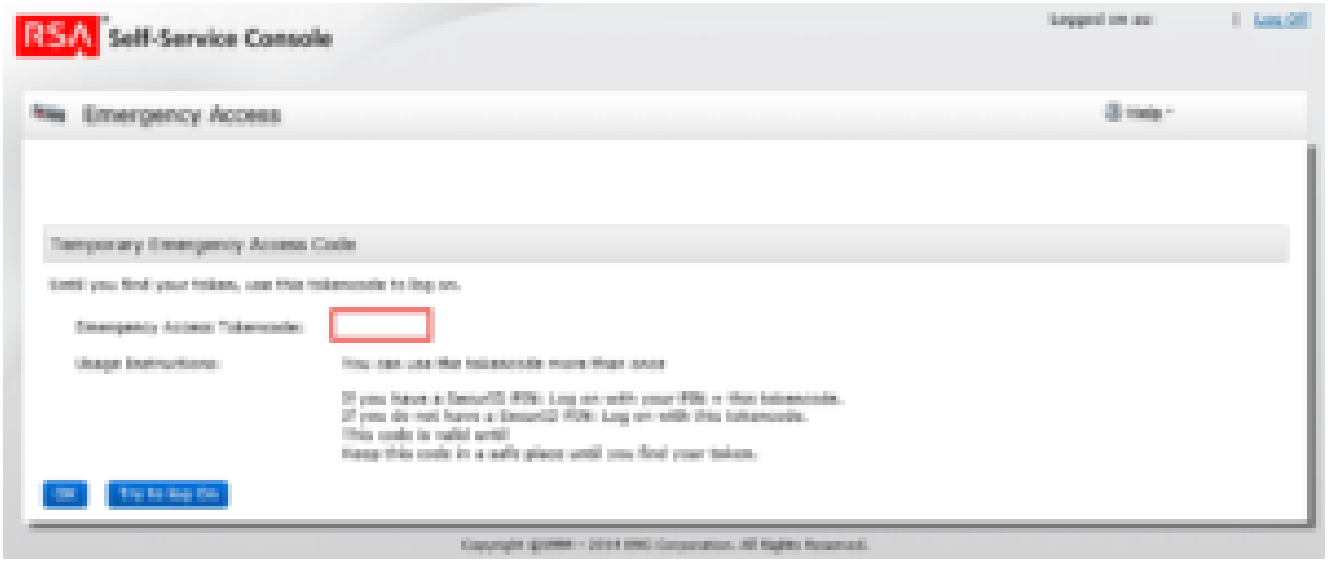
Now, fortunately to avoid a username enumeration vulnerability, RSA returns a question for every username enter, regardless if it is valid.



By default there are three Security Questions, so if you can't discover the name of their sixth grade teacher, perhaps you can find the name of their maternal grandmother's first name.

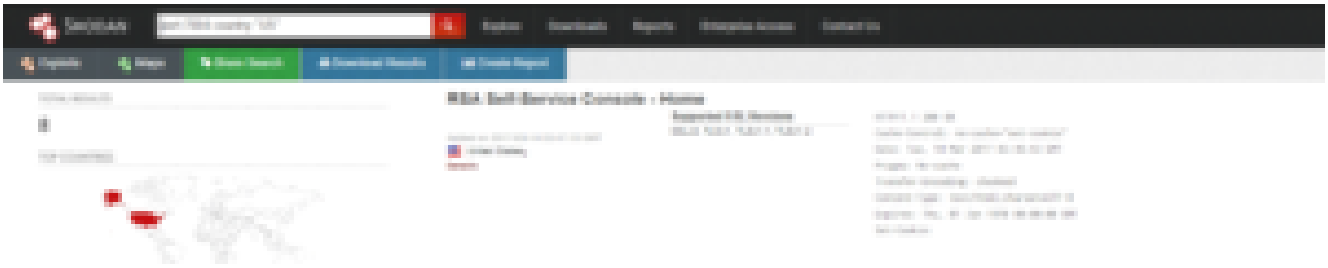
However, with some clever research, we can sometimes find the answer to the user's security question. If we get the question correct we are then brought to this familiar page.

Where, we can again be issued an Emergency Access Tokencode.



One problem that I have consistently run into, is discovering RSA servers, as they do not seem to register an SPN, nor are they consistently registered in DNS. However, they do often run on the default port of 7004, of which very little else run on as well.

… and Shodan seems to agree.



Additionally, RSA consoles have the default title “Self-Service Console – Home”, which is also searchable.

Placing the RSA console outside the security boundary they are attempting to harden has always been a risky idea. Exposing it to the internet wasn’t a good idea before, it’s an even worse one now.

So about that two factor VPN…