Breach Detection at Scale
with AWS honey tokens

DAN BOURKE & DANIEL GRZELAK | ATLASSIAN SECURITY
AWS keys
AWS root access keys.csv do not delete this is important
how to build a serverless project.jpg
A policy defines the AWS permissions that can be assigned to a user, group, role, or resource. You can construct a policy using the visual editor or create a policy document using the JSON editor.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "*",
            "Resource": "*",
            "Effect": "Deny"
        }
    ]
}
```
Canarytokens by Thinkst

What is this and why should I care?

Brought to you by Thinkst Canary, our insanely easy-to-use honeypot solution that deploys in just four minutes. Know. When it matters.

© Thinkst Applied Research 2015–2018
What if you want over 9000?

Security Advice: But less than 10,000
Security Advice: The space is in the crab.
MONITOR ALL THE THINGS!
You may have some questions
How do I set this up?
https://bitbucket.org/asecurityteam/spacecrab
Welcome to Project SPACECRAB setup.
We'll begin constructing spacecrab infrastructure shortly,
but first we'll need to collect some data.

In order to secure your SPACECRAB infrastructure, we will need to limit access to a certain user or role
You are currently authenticated as:

```
arn:aws:iam::012345678901:role/AQUASTREAM-SysAdmin
```

Would you like to use arn:aws:iam::012345678901:role/AQUASTREAM-SysAdmin as your admin role? Y/n:

We'll need a path, which will be part of the generated token's ARNs.
The path must start and end with /
Please enter a "Path" for your users (defaults to /SpaceCrab/): /SpaceCrab/

We'll use the path "/SpaceCrab/", is this ok? Y/n:

Would you like to send alerts to PagerDuty? Y/n: y

We will need a Pagerduty Events API v2 integration key for this alert to work.
Please enter a Pagerduty Events API v2 integration key: sdagjhadskjgasdkhgdgjo
Using sdagjhadskjgasdkhgdgjo for Pagerduty Integration, is this correct? Y/n:

Would you like to send alerts via email? Y/n: n

Uploading CloudFormation templates.
Uploading backup-function.template
How do I deploy tokens?
#!/bin/bash
exec 2>&1

loggedInUser=`/bin/ls -l /dev/console | /usr/bin/awk '{ print $3 }'`
if [ ! -e /Users/$loggedInUser/.aws-backup/credentials ]; then
  API_KEY=$4 # pass api key in from casper as arg 4
  URL=$5 # and the url from casper as arg 5
  DATA='{"Owner":"Security Intelligence Team", "Location":"$loggedInUser" laptop", "Notes":"created by casper"}"
  RESP=`curl -X POST -H "x-api-key: $API_KEY" -d "$DATA" $URL`
  AKID=`echo $RESP | tr -d "{}" | sed 's/.*AccessKeyId:/"\1"/; s/ */\n/"
  SAK=`echo $RESP | tr -d "{}" | sed 's/.*SecretAccessKey:/"\1"/; s/ */\n/"
  mkdir -p /Users/$loggedInUser/.aws-backup
  echo -e "[default]\naws_access_key_id = "$AKID\naws_secret_access_key = "$SAK >> /Users/$loggedInUser/.aws-backup/credentials
  chown -R $loggedInUser:staff /Users/$loggedInUser/.aws-backup/
  echo "saved token to file for $loggedInUser"
else
  echo "already run or $loggedInUser has a backup file already"
fi
Maybe those creds are honeytokens and the Hawaiian missile people are galaxy brain trolls.
Does telling people about this make it useless?

Security Advice: full disclosure c’mon what’s the worst that could happen
The concept and use of canary tokens has made me very hesitant to use credentials gained during an engagement, versus finding alternative means to an end goal. If the aim is to increase the time taken for attackers, canary tokens work well.
How much does it cost?
What I Learned
<table>
<thead>
<tr>
<th>Date</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>30/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>28/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>27/12/17</td>
<td>+1 (206) 266-1000</td>
<td>Seattle, WA, United States</td>
</tr>
<tr>
<td>24/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>24/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>24/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>23/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
<tr>
<td>22/12/17</td>
<td>Mobile 206-1000</td>
<td></td>
</tr>
</tbody>
</table>
Hi Dan,

It was a pleasure speaking with you today.

Unfortunately, I must ask you to delete all access keys that are exposed. The easiest way to delete them all (as most are still listed as active) is to close your account.

This action is directly related to the AWS Acceptable Use Policy I mentioned on our phone call.


I'm sorry for any inconvenience this may cause, but I must ask you to comply as soon as possible.

Once this is done, please write me back using the link below.
Security Advice: Don’t be Goofus.

Goofus and Gallant

By Garry Cleveland Myers
Pictures by Marion Hull Hammel

Goofus runs with the scissors pointing up.

Gallant walks with the scissors pointing down.
### SpaceCrab Statistics

<table>
<thead>
<tr>
<th>Github/Gist</th>
<th>Exploitation %</th>
<th>Avg Time-To-Exploit</th>
<th>Fastest TTE</th>
<th>Slowest TTE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>82.38% (201/244)</td>
<td>30m 8s</td>
<td>30m 4s</td>
<td>30m 15s</td>
</tr>
</tbody>
</table>
# SpaceCrab Statistics

<table>
<thead>
<tr>
<th>Exploitation %</th>
<th>Avg Time-To-Exploit</th>
<th>Fastest TTE</th>
<th>Slowest TTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pastebin</td>
<td>9.33% (7/75)</td>
<td>24h 49m 16s</td>
<td>12h 45m 22s</td>
</tr>
</tbody>
</table>
It's been a long year building unhackable infrastructure with the team, so it's time for a well-deserved break!

```c
make_empty_array(&empty_array, a_readonly);
if (dict_find_string(op, "Filter", &pFilter) > 0) {
    if (is_array(pFilter)) {
        if (_is_array(pFilter, t_name))
            return_error(gs_error_typecheck);
        make_array(&filter1_array, a_readonly, 1, pFilter);
        pFilter = &filter1_array;
    } else
        pFilter = &empty_array;
    /* If Filter is undefined, ignore DecodeParms. */
    if (pFilter != &empty_array &&
        dict_find_string(op, "DecodeParms", &pDecodeParms) > 0 ) {
        if (pFilter == &filter1_array) {
            make_array(&parms1_array, a_readonly, 1, pDecodeParms);
            pDecodeParms = &parms1_array;
        } else if (!(_is_array(pDecodeParms))
            return_error(gs_error_typecheck);
        else if (r_size(pFilter) != r_size(pDecodeParms))
            return_error(gs_error_rangecheck);
    } else
        pDecodeParms = 0;
    for (i = 0; i < r_size(pFilter); ++i) {
        ref y, f_name, dp;
        array_get(i, pFilter, (long)i, &f);
        if (!is_array(&f, t_name))
            return_error(gs_error_typecheck);
        ref name_string_ref(i, f_name, &f_name);
        if (r_size(&f_name) <= 0)
            return_error(gs_error_rangecheck);
        if (pDecodeParms) {
            array_get(i, pDecodeParms, (long)i, &dp);
            if (!(_has_type(&dp, t_dictionary) || _has_type(&dp, t_null)))
                return_error(gs_error_typecheck);
        }
    }
```
strong finish/call to action

https://bitbucket.org/asecurityteam/spacecrab
Thanks!