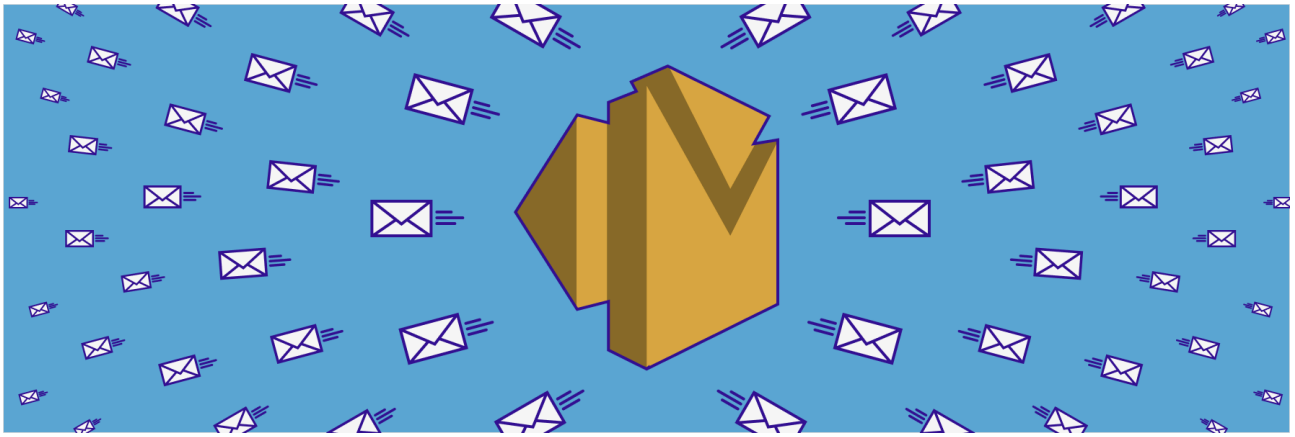
Permiso | Blog | SES-pionage

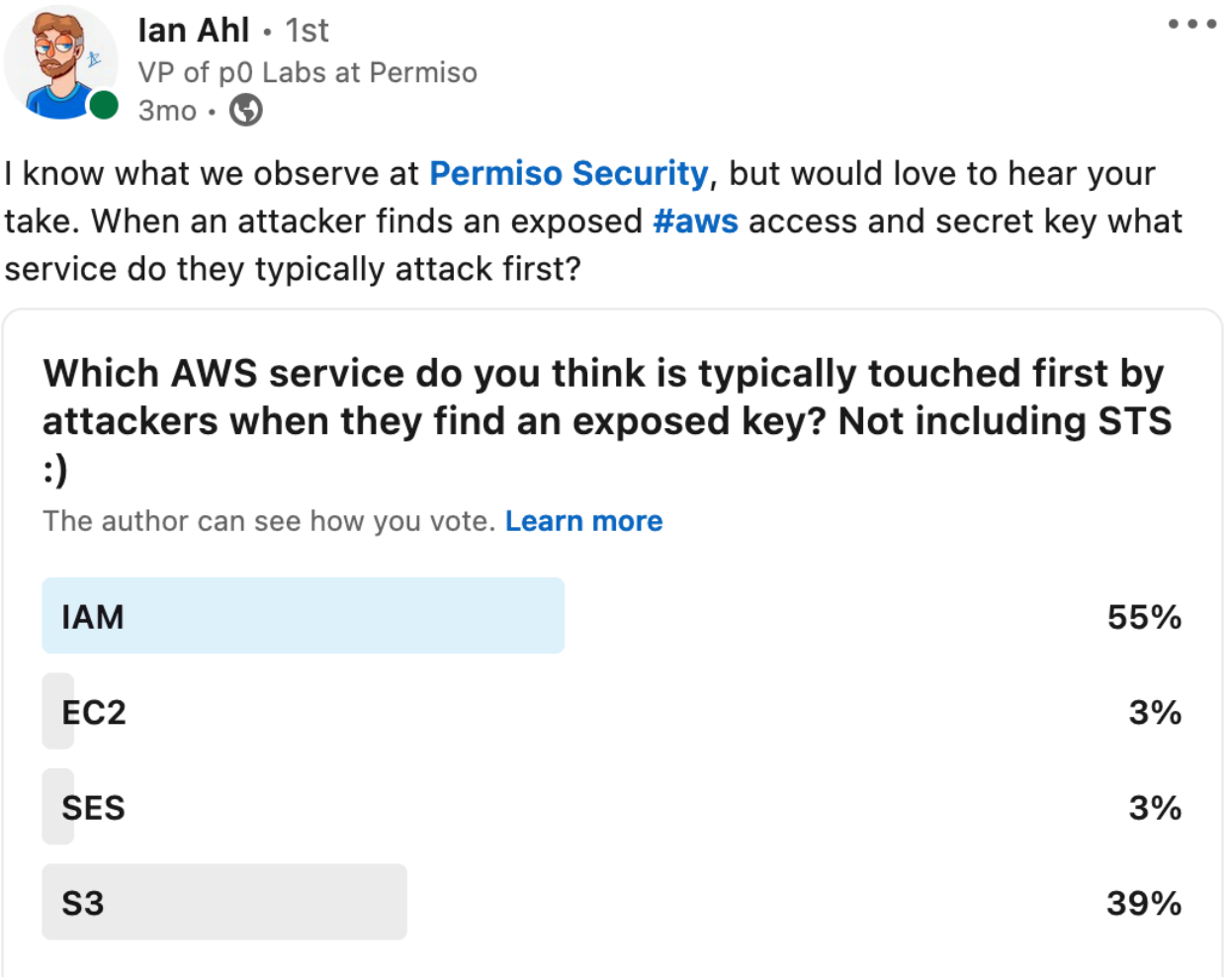
By Nathan Eades

Published: 2023-01-12 · Archived: 2026-04-05 13:53:59 UTC



Credits: Wilma Miranda

Exposed and stolen long lived keys are often the source of incidents we identify in our client's cloud environments. Detecting the abuse of these keys is an area we focus a lot of effort in. After seeing many of these incidents, patterns are starting to arise that are somewhat unexpected. We took a poll a few months back to see what service people think is typically first to be targeted by attackers. While most folks guessed IAM, our data shows that AWS Simple Email Service (SES) is usually hit first. We will use this article to talk through what SES is, what we have observed, and what you should be on the lookout for!



Initial Investigation

In October of 2022, Permiso was investigating an alert which was traced back to an exposed long-lived AWS credential or access key (AKIA***) along with its secret key on a public GitHub repository. While we had many findings, of particular interest was the following set of SES service actions:

1. `GetAccount`
2. `ListServiceQuotas`
3. `ListIdentities`

SES 101

Many, if not most organizations, especially those that provide a service or application, have their trusty email relays (SMTP relay), included as an extension of their email infrastructure. Relays are often used to send automated emails, such as those for mass marketing messages and applications, separate from internal email communications. These systems can be difficult to build and maintain and may not be cost effective.

So is AWS SES an SMTP relay? It is, and more! In short, AWS SES provides the capability of bulk email sending, this includes an SMTP interface as well as API, CLI and SDK options.

Why then does an attacker want to target SES? This is for the same reason attackers have always targeted email systems. Use of phishing and other malicious messages to spread and continue to gain access through a potentially trusted entity. Additionally, to the attacker, it's free, as you the compromised organization now get to float the bill for all that activity. In the case of SES it can also be harder on the defensive side to block the activity. See Appendix: "SES Email Defense" for more on working to block potential malicious activity originating from SES.

SES Footprinting

Due to the potential for abuse, AWS SES has several built in protections. For starters, you can't just log into AWS, pop over to the portal and start sending emails to everyone. When SES is first spun up, the account is placed into a "sandbox" environment. AWS maintains a FAQ on what it means to be placed in sandbox: https://docs.aws.amazon.com/ses/latest/dg/request-production-access.html.

- You can only send mail to

  verified email addresses and domains

  , or to the Amazon SES mailbox simulator.

- You can send a maximum of 200 messages per 24-hour period.
- You can send a maximum of 1 message per second.
- For sending authorization, neither you nor the delegate sender can send email to non-verified email addresses.
- For account-level suppression, bulk actions and SES API calls related to suppression list management are disabled.

An account outside of the sandbox environment is not held back by the same rate limits and can send emails to any address while still limited to using "verified senders".

All of this is to say that attackers know about these constraints as well and they have means provided by AWS (for legitimate means) to check for them. With that, we are back to the discussed SES service actions; `GetAccount`, `ListServiceQuotas`, `ListIdentities`, and for sake of argument were going to add 4 more, `GetSendQuota`, `GetIdentityVerificationAttributes`, `UpdateAccountSendingEnabled` and

`GetAccountSendingEnabled`.

Breaking the commands down into categories:

Verified Sender	Sandbox / Quota	Account SES Sending Status
ListIdentities GetIdentityVerificationAttributes	ListServiceQuotas GetSendQuota GetAccount	UpdateAccountSendingEnabled GetAccountSendingEnabled

The use of `ListIdentities` is meant to provide the attacker with knowledge of how SES senders may be configured within an account. As previously described, a verified domain is more valuable to an attacker as this provides endless options for usernames and can make blocking the emails more difficult.

Example `ListIdentities` Response:

```
{
   "email": [
       "example@example.com",
       "example2@example.com",
       "example@gmail.com"
   ],
   "domain": [
       "test-permisosec.awsapps.com",
       "other-test-permisosec.example.com"
   ]
}
```

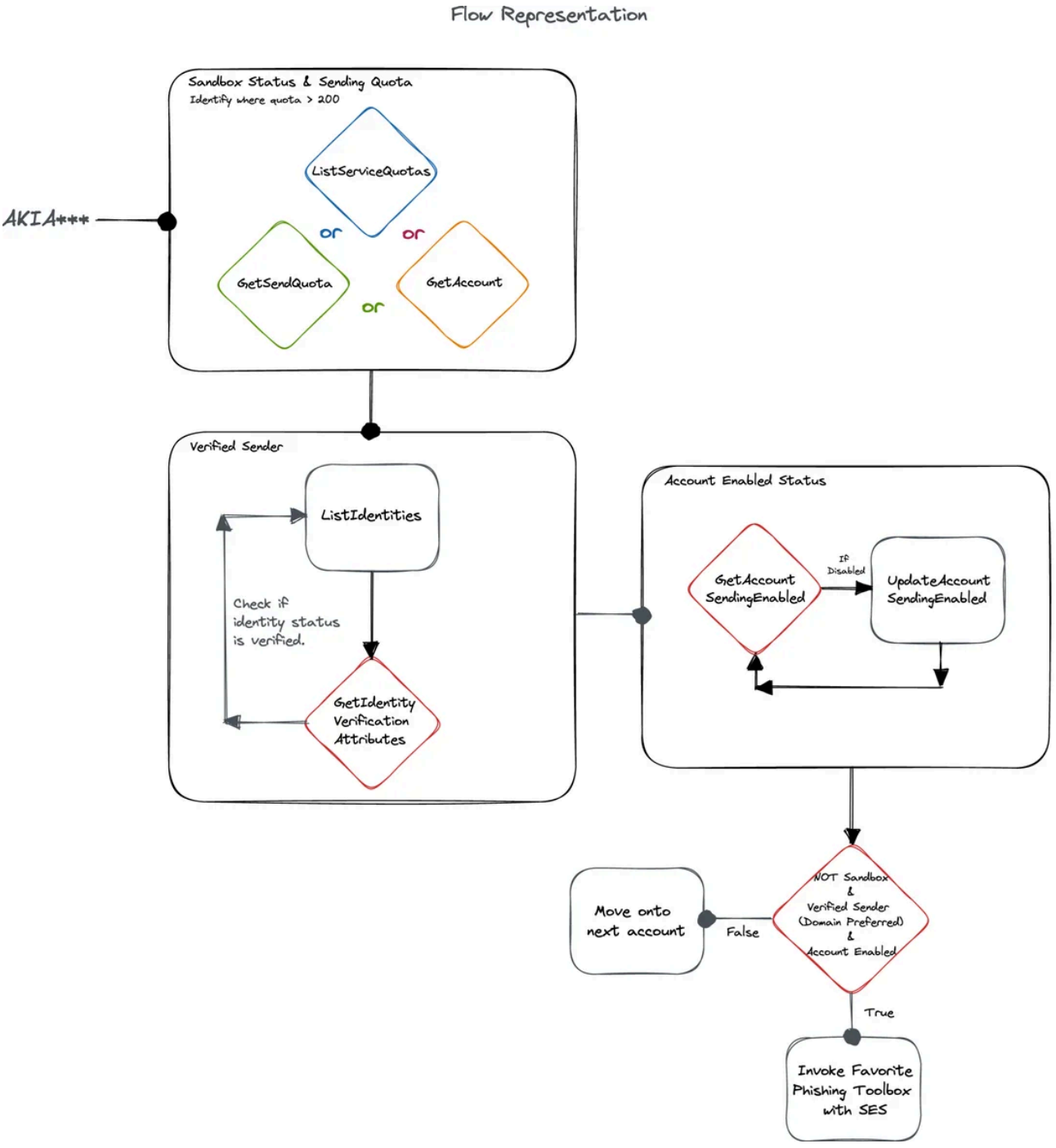
Note that `ListIdentites` does not provide a verification status. The response includes any identity regardless of verified status and each has no guarantee to be verified. `GetIdentityVerificationAttributes` provides this final piece, providing a status for each identity and allowing an attacker to confirm if there are any verified senders.

`ListServiceQuotas`, `GetSendQuota` and `GetAccount` may each be utilized to provide the accounts send quota for the SES mailing service. Though what truly matters to the attacker is if the response is not **200**. A response of 200 indicates the service is in sandbox mode. The account cannot send to any address that is not verified and is limited to only 200 messages over 24 hours.

Finally, `UpdateAccountSendingEnabled` and `GetAccountSendingEnabled` have the intent to ensure the compromised account is capable of sending messages.

Detection

Our research has found that SES footprinting has been well established and today follows a few typical flows.



Most SES abuse we have observed originates from a long lived access key (AKIA***). Looking for the following set of actions from long lived keys will serve as good signal of SES abuse.

1. Sandbox Status enumeration: Using one or many of `ListServiceQuotas`, `GetSendQuota` and `GetAccount` allows the attacker to reach a reasonable conclusion on whether the account is still in sandbox. Though keep in mind, even in sandbox, this could be means to spread internally within an organization through verified addresses / domains.

2. Verified Sender enumeration: `ListIdentities` will be utilized to gather a list of domains and emails, sometimes in combination with `GetIdentityVerificationAttributes` to retrieve the list entries verification status.

3. Account Enablement enumeration/modification: Attackers typically leverage `UpdateAccountSendingEnabled` to verify and set, if they have the privilege, the account sending status. We also want to check for an advanced attacker whom may avoid using a “Write” action if it’s not necessary through the `GetAccountSendingEnabled` action.

For Permiso customers, the following alerts are indicative of SES targeting and abuse:

ID	**Description**
P0_AWS_SES_ACCESSKEY_GET_ACCOUNT_1	AWS Simple Email Service (SES) Get Account performed by long term access key. SES GetAccount may be utilized to obtain the accounts sending quota to check for a sandbox environment. This may signal that the organization is being tested or targeted for use in SES mailing attacks.
P0_AWS_SES_ACCESSKEY_QUOTA_DETAILS_1	AWS Simple Email Service (SES) GetSendQuota or ListServiceQuotas of the SES service, performed by long term access key. Checking SES quotas is indicative of testing for an SES sandbox environment, often performed by attackers in preparation for an SES mailing attack.
P0_AWS_SES_ACCESSKEY_LIST_IDENTITIES_1	AWS Simple Email Service (SES) List Identities performed by long term access key. SES ListIdentities retrieves available email addresses and domains within the account. This may signal that the organization is being tested or targeted for use in SES mailing attacks.

P0_AWS_SES_ACCESSKEY_LIST_IDENTITIES_VERIFIED_1	AWS Simple Email Service (SES) List Identities with GetIdentityVerificationAttributes performed by long term access key. SES ListIdentities with GetIdentityVerificationAttributes retrieves available email addresses and domains within the account and provides their status, such as “verified”. This may signal that the organization is being tested or targeted for use in SES mailing attacks.
P0_AWS_SES_ACCESSKEY_VERIFY_SENDING_STATUS_1	AWS Simple Email Service (SES) GetAccountSendingEnabled performed by long term access key. SES GetAccountSendingEnabled provides the disabled or enabled status of the account. This may signal an attacker checking the current status in preparation for use in SES mailing attacks.
P0_AWS_SES_ACCESSKEY_ENABLE_SENDING_1	AWS Simple Email Service (SES) UpdateAccountSendingEnabled performed by long term access key. SES UpdateAccountSendingEnabled allows for updating the status. In this case, set to “enabled”. This may signal an attacker ensuring the use of the account in preparation for SES mailing attacks.
P0_AWS_SES_ACCESSKEY_ABUSE_1	The activity observed is a combination of events, these events when observed on their own are considered to be abnormal AWS Simple Email Service

	(SES) activity. When observed together the events follow a well known set of TTPs seen before an account is utilized in SES abuse.

Appendix:

SES Email Defense:

Lets say the ***header from*** shows as `test@test-permisosec.awsapps.com`, pending the SES setup, the attacker may be able to use any username section of the email address e.g. `test2@testing.test-permisosec.awsapps.com`. This means you are forced to block at the domain level, of course if the compromised organization is heavily utilized under legitimate circumstance then this presents a business problem. The *envelope from,* `010001858f7f9534-7ef4928a-f757-455f-9c33-959c7b98864f-000000@[amazonses.com](<http://amazonses.com>)` won't provide any benefit as the username field is a random string and part of the `amazonses.com` domain used by any organization doing business through the AWS SES service.

From the receiving side, one of your best courses of action is to notify your user base and to contact AWS: abuse@amazonaws.com - "Provide all the necessary information, including logs in plaintext, email headers, and so on, when you submit your request."

- https://aws.amazon.com/premiumsupport/knowledge-center/report-aws-abuse/
- https://support.aws.amazon.com/#/contacts/report-abuse

Pending the compromised organization and readily available abuse points of contact, providing the same data to them may be beneficial as well.

SES Verification Methods:

Domain / Email Verification: Even outside of SES sandbox, a sender must be verified. This again is a great control from AWS to ensure the account owner is in control of the domain or email address being used. Verifying a single email address requires acknowledging receipt of the message from the inbox. Verifying a domain requires DNS control of the domain as full DKIM verification will be required.
https://docs.aws.amazon.com/ses/latest/dg/creating-identities.html

An example of single email address verification:

Congratulations!

You have successfully verified an email address. You can now start sending email from this address.

For new Amazon SES users—If you have not yet applied for a sending limit increase, then you are still in the sandbox environment, and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Identity Management** section of the Amazon SES console.

For new Amazon Pinpoint users—If you have not yet applied for a sending limit increase, then you are still in the sandbox environment, and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Settings** > **Channels** page on the Amazon Pinpoint console.

If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

Thank you for using Amazon Web Services!
