

United States Senate

WASHINGTON, DC 20510

October 24, 2019

The Honorable Joseph J. Simons
Chairman
Federal Trade Commission
600 Pennsylvania Avenue, NW
Washington, DC 20580

Dear Chairman Simons:

We write to urge the Federal Trade Commission (FTC) to open an investigation to determine if Amazon's failure to secure the servers it rented to Capital One may have violated federal law.

On July 29th, 2019, Capital One revealed that a hacker had breached its systems and stolen the personal data of 100 million Americans. As Amazon acknowledged in the attached August 13, 2019 letter, the hacker stole data from Amazon servers rented by Capital One using a hacking technique known as a "server side request forgery (SSRF) attack."

SSRF attacks can be used by hackers to steal valuable data from servers rented from cloud computing companies. Amazon's largest competitors have included mandatory protections against SSRF attacks in their products for several years — Google since 2013 and Microsoft since 2017. Amazon's failure to add a similar software protection against SSRF attacks to its Amazon Web Services (AWS) cloud computing product has been the subject of significant public discussion among cybersecurity experts for the past five years, including in presentations at major industry conferences.

As you know, the FTC has made it clear that companies have an obligation to act on third-party reports of cybersecurity vulnerabilities. In the FTC's 2013 case against the smartphone manufacturer HTC, the FTC established that companies must "implement a process for receiving and addressing security vulnerability reports from third-party researchers, academics or other members of the public." HTC's failure to do so, the FTC argued then, constituted an unfair business practice.

While it is likely that Amazon has known that its AWS product was vulnerable to SSRF attacks since the first high-profile demonstration by a cybersecurity researcher in 2014, the company has certainly known since mid-2018 at the latest. In August of 2018, Amazon's security team was contacted by email by a cybersecurity expert, who recommended that Amazon adopt the same cybersecurity defense against SSRF attacks already used by Google and Microsoft. A copy of that email is attached. Amazon failed to act on this third-party report and has not provided an explanation for its inaction.

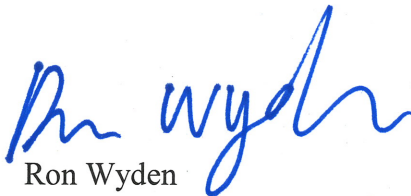
Amazon knew, or should have known, that AWS was vulnerable to SSRF attacks. Although Amazon's competitors addressed the threat of SSRF attacks several years ago, Amazon continues to sell defective cloud computing services to businesses, government agencies, and to

the general public. As such, Amazon shares some responsibility for the theft of data on 100 million Capital One customers.

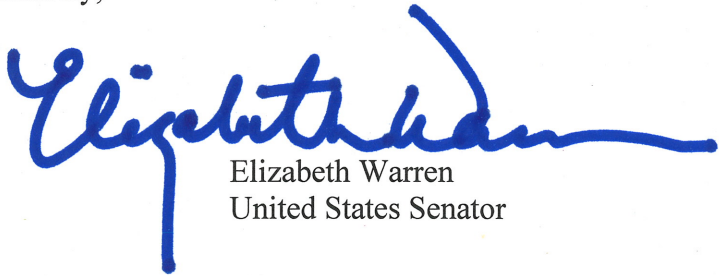
The FTC has the authority and responsibility to investigate unfair and deceptive business practices. We urge you to investigate whether Amazon's failure to secure its services against SSRF attacks constitutes an unfair business practice, which would violate Section 5 of the FTC Act.

Thank you for your prompt attention to this important matter.

Sincerely,



Ron Wyden
United States Senator



Elizabeth Warren
United States Senator



August 13, 2019

The Honorable Ron Wyden
United States Senate
221 Dirksen Senate Office Bldg.
Washington, D.C., 20510

Dear Senator Wyden,

Thank you for your letter of August 5, 2019. We are happy to answer your questions – as well as provide some additional context.

First, your letter points out that because so many companies use Amazon Web Services (“AWS”), the security of AWS’s services is critical. We could not agree more. Security is our number one priority at AWS by far, and we care deeply about both our services and our customers’ data being secure. We invest a substantial amount of resources securing our services and helping our customers secure themselves, and will continue to do so forever. Most enterprise and government customers who consider AWS thoroughly inspect our security architecture, services, and practices – with frequent deep security and engineering conversations between technical leaders at AWS and these customers. We consistently hear from customers that moving to the Cloud and AWS is helping them have a stronger security posture than their prior on-premises footprint. The security strength of AWS is one of the primary reasons our business has grown as fast as it has.

Regarding your specific questions:

Your first question asks about cybersecurity experts publicly speculating that the person implicated in the Capital One incident exploited a “Server-Side Request Forgery (SSRF) vulnerability” and asks whether, to the best of our knowledge, a SSRF attack was used to gain access. As Capital One outlined in their public announcement, the attack occurred due to a misconfiguration error at the application layer of a firewall installed by Capital One, exacerbated by permissions set by Capital One that were likely broader than intended. After gaining access through the misconfigured firewall and having broader permissions to access resources, we believe a SSRF attack was used (which is one of several ways an attacker could have potentially gotten access to data once they got in through the misconfigured firewall).

Your second question asks about the number of AWS customers that have been compromised through SSRF attacks and how many of those attacks involved our metadata service. As discussed above, SSRF was not the primary factor in the attack. We are not aware of any other noteworthy SSRF compromises of AWS customers. It’s possible that there have been small numbers of these that haven’t been escalated to us (we have millions of active customers using our services every month), but none that we have confirmed at any significant scale, beyond

Capital One. We understand that the person implicated in the Capital One attack identified several other organizations that she believed she had successfully attacked in some form. We quickly reached out to those customers to make them aware of those claims, and then to offered to help them assess and secure their data. To date, these customers have not reported any significant issues.

Your third question asks what guidance, if any, AWS provides to customers about SSRF attacks, particularly against our metadata service. Most people who know security will tell you that the best way to ensure strong security is to have multiple layers of protection with intentional redundancies, as this creates “defense in depth.” The web application firewall (“WAF”) is the first layer and serves as the “front door” to a customer’s resources; if not configured properly, the WAF may enable attackers to access resources they should not be able to access. We give customers clear guidance on both the importance and necessity of protecting themselves from SSRF attacks, as well as other attack vectors. We provide documentation, how-to-guides, and professional services to help customers set up WAF protections. We also offer our own AWS Web Application Firewall (“AWS WAF”), which has expansive capabilities through which customers can completely block SSRF and other attacks.

We offer the same guidance and tools to help customers set up the right permissions for their resources, which is the next stage of protection after the WAF. Even if a customer misconfigures a resource, if the customer properly implements a “least privilege” policy, there is relatively little an actor has access to once they are authenticated — significantly diminishing the customer’s risk. We also offer detection services that add another layer our customers can deploy to protect their resources. We have a service called Macie that automatically classifies data into different buckets of sensitivity, and then sends customers alarms when either an anomalous requester tries to access objects or if there is an unusually high volume of data being moved. We have a service called GuardDuty that alerts customers when there are unusual Application Programming Interface (“API”) calls. We also have a service we provide called a “Well Architected Review,” where we inspect a customer’s technology architecture, and give feedback on whether we believe that customer is well-architected according to best practices. These are just a few of the many security practices and layers that we provide customers. We have not called out the instance metadata service directly, and that’s because it is one of several sub-systems deep in the technology stack that is at the tail-end of a lot of security layers customers can deploy to protect themselves. While we already offer protections related to the instance metadata service, we are going to add an additional protection in the near future to provide even more protection for our customers.

Your fourth question asks about a Netflix request to add a header to protect the metadata service from SSRF attacks. Netflix effectively runs all of their applications on AWS, and as such, we have an expansive relationship with Netflix that spans dozens of people, scores of feature requests, and hundreds (maybe thousands) of conversations a year. Our relevant product leaders were not aware of that request from Netflix, and Netflix has said both that this engineer’s tweet does not reflect their views and that “Netflix has no technical issues with Amazon.”

Capital One is a sophisticated and thoughtful company, with excellent technology and security organizations. Sometimes humans make mistakes. And, while the Capital One attack happened

due to the application misconfiguration mentioned above, there are several actions AWS will take to better help our customers ensure their own security. First, we will proactively scan the public IP space for our customers' firewall resources to try and assess whether they may have misconfigurations. We started doing so last week, and we will notify customers proactively of any firewall resources we think could be misconfigured. We will not be able to definitively know whether a firewall is misconfigured (only the customer truly knows what they intended with resources under their control), but if we think there might be a misconfiguration, we will err on the side of over-communicating. Second, we will redouble our efforts to help customers set the least permissive permissions possible. Third, we will push harder to make our anomaly detection services (Macie and GuardDuty) more broadly adopted and accessible in every geographic region in which we operate. We will look at additional "belt and suspenders" we can add to sub-systems deeper in our stack (like the instance metadata service) to provide even more protection for customers. Security will always continue to evolve at a rapid pace, and we will surely find other areas we can improve moving forward. But, you can rest assured that we will learn from this event alongside our partner, and be relentless in continuing to evolve our services over time.

Thank you for the invitation to have a dialogue on these critical issues. Our teams working on this issue welcome the opportunity for further discussion. Please feel free to contact me if I can provide any further information or be of any further assistance.

Sincerely,



Stephen Schmidt
Vice President, Chief Information Security Officer
Amazon Web Services



Feature request: Provide a way to better protect the metadata service

3 messages

To: aws-security@amazon.com

Tue, Aug 28, 2018 at 8:43 PM

With the recently disclosed issue here: <https://twitter.com/disclosedh1/status/1034138522641424384> and with this having been an issue for a long time (I even included it part of the flaws.cloud security training a year and a half ago), steps should be taken to better secure the AWS metadata service. Specifically, when an EC2 accesses 169.254.169.254 in order to get its session tokens, it should have to provide a host header like Google's GCP service requires for its metadata service. See the mention of "Metadata-Flavor: Google" on <https://cloud.google.com/compute/docs/storing-retrieving-metadata>

Doing that would help prevent SSRF issues and others where session credentials are stolen.

It would also be helpful to be able to lock down the credentials so that they can only be used on the EC2 they were provided to. I'm not sure on the best way to accomplish that though, but likely some sort of IAM variable that represents the external IPs of the account, so at least the EC2 creds would be locked down to that account.



aws-security@amazon.com <aws-security@amazon.com>

Tue, Aug 28, 2018 at 11:03 PM

Cc: aws-security@amazon.com

Hi [redacted]

Thank you for bringing your security concern to our attention. We greatly appreciate and encourage reports from the security community worldwide.

We will route this feedback to the relevant internal team for consideration.

If you discover or become aware of other security concern specific to AWS products and services, please do not hesitate to contact us again at aws-security@amazon.com

Damian J.
AWS Security
<https://aws.amazon.com/security>
[Quoted text hidden]

To: aws-security@amazon.com

Tue, Sep 4, 2018 at 8:29 AM

FYI, yet another example of the metadata service being abused: <https://labs.mwrinfosecurity.com/blog/from-http-referer-to-aws-security-credentials/>

