Wesley Kirkland

My first vulnerability – Mimecast Sender Address verification

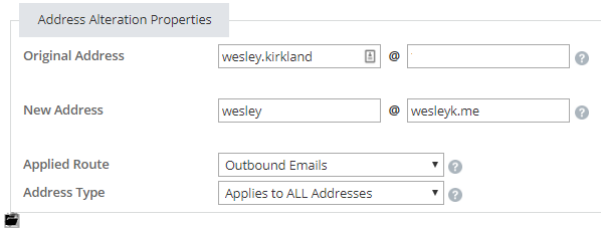
I want to start and say this was something completely new to me, my only other previous security research experience was something a coworker noticed and I helped report.

So what went wrong, to me the backstory is pretty interesting. A little over a year ago this (https://www.reddit.com/r/sysadmin/comments/894q5n/i_just_violated_tos_of_our_enterprise_mail/) was posted on Reddit. In short, the poster violated their mail providers Terms of Service by sending mail as another customer. While yes it shouldn't have been done it was fairly harmless in the grand scheme of things. Though this kind of independent third-party testing and disclosure, while the poster did not appear to do a responsible disclosure. They were not encouraged to by the vendor, as the vendor just said our policies should have prevented this instead of a technical solution. At the time this was released I was getting pretty heavy with email architecture and email security.

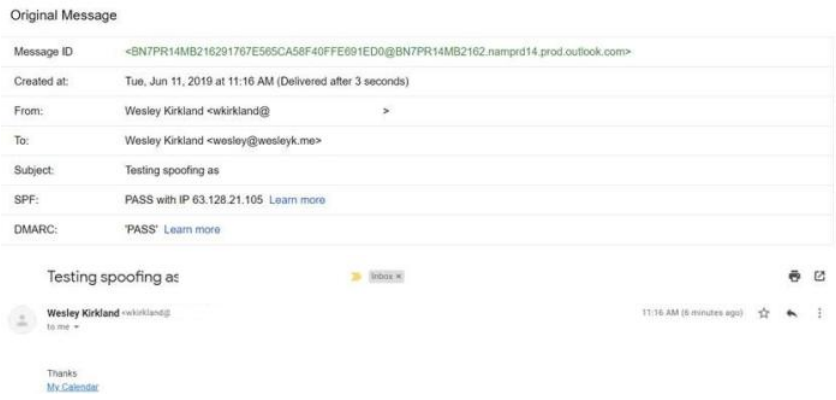
My employer has a unique need to be able to send as multiple primary email addresses. This is something we had talked about before we ever purchased Mimecast, as I had heard about how Cracker Barrel uses Mimecast to do address rewrites. While this may not sound unique to everyone reading this, to me it was a first and I need to find a way to do it at scale without significant costs while creating easy integrations. About a year after we purchased and implemented Mimecast I was getting more into this requirement and decided to try out the address alternation rewrite functionality. In this demo, we had domainb.com in another Mimecast account than our primary and I was doing a demo on the fly. So during the demo, we changed the RFC5322.From and everything passed as far as SPF went, and the user was super happy! Also during that demo, we tested sending a calendar through the address rewrite and that failed, I ended up doing some more digging and noticed this was a bug. Please note that this is still not fixed and active (Ticket opened 3/12/2019), under this we learned that calendar invites were not being rewritten and failing back to our primary domain and didn't pass DMARC.

Spoofing

Sadly before when this happened I wrote it off and didn't think too much about the fact that I was able to send mail as domainb.com when I was in the Mimecast account for domaina.com. Now on June 10th (6/10/19) I did some more testing and was curious about how far I could extend this. To start I was curious if I could spoof my own personal domain of wesleyk.me. To do this is actually quite simple from within the Mimecast portal. After I added it then I just had to send mail as me to my personal domain. Do note that this failed as wesleyk.me is protected by DMARC and GSuite observed that, though it pointed out an issue that Mimecast was allowing me to send mail as domains that I didn't own.



My next test for further exploitation was more evil but I wanted to see what could be done. Knowing how email works makes testing this very simple, we can start by knowing the default MX records for Mimecast are us-smtp-inbound-1.mimecast.com, and us-smtp-inbound-2.mimecast.com. Now we can work on finding a domain that has these same MX records so we can determine who is a customer. This is easier than a bunch of misc searching thanks to one of my favorite websites. ViewDNS.info (https://viewdns.info) has an easy search function for a reverse MX lookup that we can run like this (https://viewdns.info/reversemx/?mx=us-smtp-inbound-1.mimecast.com). Please note that DNSTrails can also do this as shown here (https://securitytrails.com/list/mx/us-smtp-inbound-1.mimecast.com). From a list like this, we can easily export the data and start sending as other customers. Once we've tested sending as other customers we can take it another step further and find large brands that utilize Mimecast for Inbound and Outbound email, as well as have a p=reject DMARC policy. From doing a little research using all public information we can find brands such as Zendesk.com (https://mxtoolbox.com/SuperTool.aspx?action=dmarc%3aZendesk.com&run=toolpage), and MissUniverse.com (https://mxtoolbox.com/SuperTool.aspx?action=dmarc%3aMissUniverse.com&run=toolpage) and see what damage could be done. Neither of these brands were used during my testing, instead of finding a major high profile brand. I simply exported the list from viewdns.info, and wrote a quick script to go and find domains that had an MX record for Mimecast and a p=reject DMARC record. This took only about 10 or so minutes and we were in business. From this point, we know that we can spoof non-customers and customers, though can we spoof customers and pass DMARC? By using our address alteration policy above we can change our original email address to one of these domains. I've edited out the domains that I spoofed for privacy reasons. At this point, I disclosed the vulnerability to Mimecast through a support ticket on 06/11/19



From the images above we can see a few interesting things.

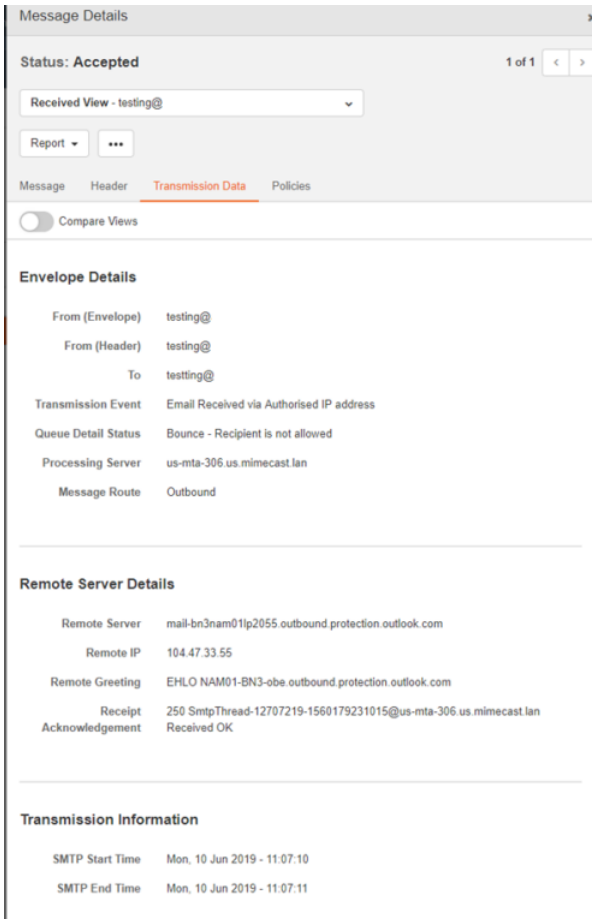
1. The IP of 63.128.21.105 (https://whois.arin.net/rest/net/NET-63-128-21-0-1/pft?s=63.128.21.105) belongs to Mimecast so we know Mimecast send the mail
2. SPF passed, we know this due to the RFC5321.From aligning to the domain via this SPF record

    1. `v=spf1 include:_netblocks.mimecast.com ~all`

3. DMARC passed. this is what makes this severe. DMARC is supposed to be one of the end all email security protocols to where you explicitly authorize a sender to spoof your domain for legitimate purposes.

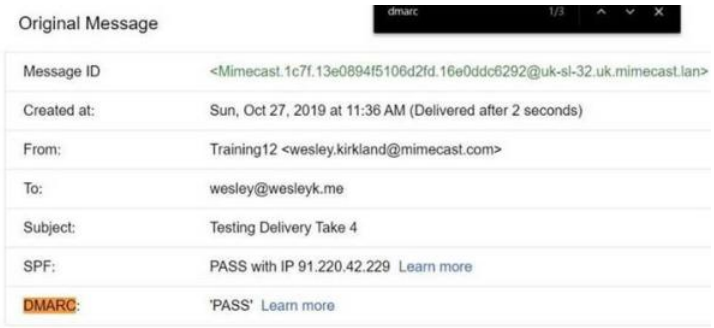
Now what we can learn from this? Since we're able to pass DMARC via the SPF mechanism, anybody that is the first person to receive our mail will have it automatically pass all authentication checks. As well as any mail that is a forward and keeps passing through Mimecast's' servers will be authenticated even though they're exploding the message as it comes from their authorized IPs. We can't pass DKIM checks due to we don't have control of the DNS zone. We can prevent DMARC failures simply by never DKIM signing the mail once it leaves our environment.

We can investigate why this happened even further, if you look at the Transmission event you will see “Email Received via Authorized IP address”. This occurred due to how the Mimecast platform works, their own IP addresses are authorized IP addresses since the platform can be used to send an email when your backend mail server is down. Due to the lack of proper checks for if you’re allowed to send as any domain has caused some significant issues for them.

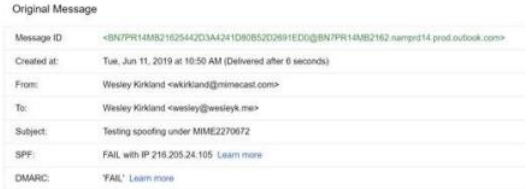


Further Exploitation

On October 28th I was fortunate to be able to attend Mimecasts’ first security conference down in Dallas, TX. At their conference, they had training going on, and we all had access to their mimecasteducation.com training tenant. Now, this can get fun, if we do some MX/SPF record lookups (https://mxtoolbox.com/SuperTool.aspx?action=spf%3amimecast.com&run=toolpage) we can see that mimecast.com is hosted in their EU grid. We can also determine that mimecasteducation.com (https://mxtoolbox.com/SuperTool.aspx?action=spf:mimecasteducation.com&newAppVersion=1) is hosted there as well. Knowing this information and knowing that the SPF record is there, I was able to successfully do an address alternation and pass DMARC for Mimecast.com.



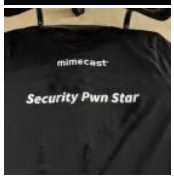
During my initial testing, I tried this and partially failed as I didn’t have access to their EU grid. Though both of these are highly worrying as I was able to successfully send emails as their primary brand and get it delivered.



Note: Since we’re not able to run a MITM attack or hijack the domain/DNS we can only send mail. Though if we want to be tricky we can set the reply to (RFC5322.Reply-To) address in the mail field to an email address that we control, or even register a similar domain. Then make our email on the left half of the email address the same, assuming the user falls for our initial spoof we can successfully continue phishing and expand into social engineering. Since we’re able to pass email authentication checks. All of this should be successful in theory as DMARC does not verify the RFC5322.Reply-To header.

Timeline

- 06/11/2019 – Initial disclosure date
- 06/24/2019 – No follow up on my ticket, so I contacted my Customer Support Manager and sales rep to escalate this. At this point, it had been 13 days since disclosure with no acknowledgment of the issue.
- 06/25/2019 – I received an internal email forward stating this
  - REDACTED and I have been in communication with the internal disclosures team since the time of Wesleys report. We had a conference call with product on Friday and expect to have a call arranged with the customer early this week. Please let REDACTED or I know if you have any questions.
  - This is good to know that they were in communication with Engineering, I looked up the name of the person who sent the above response. They're a manager of support at Mimecast
- 06/28/2019 – I had a call with their engineering/security team on that they've confirmed what's going and they're working to fix it. I was notified at this point it would take about one month to fix
- 08/06/2019 – No response or fix from the previous call, so I reached out to my contacts again
- 08/12/2019 – No response so I emailed my original Mimecast ticket to their disclosures distribution list with this context
  - Mimecast Disclosure Team,

    Can you please provide an update on the disclose submitted under REDACTED. It's been 1.2 months since my last call with Mimecast, and on that call it was estimated to have it fixed within 1 month.
- 10/11/2019 – Still no confirmation or timeline of it being fixed, I contacted my contacts again to try and get some traction
- 10/28/2019 – Successful spoof of Mimecast.com
  - My account manager emails me the internal code name for the project, this is wonderful news as I'm at the Mimecast conference. I was able to get a few technical details but not the explanation behind them.
  - In more fun news I was wearing my Disclosures' T-Shirt from a previous Mimecast Disclosure that didn't fully classify
  - 
    
    

- 10/29/2019 – I attended one of the few technical deep-dive sessions which was fantastic by the way. At the end of it, I got a sneaking suspicion that this person might be able to give me some more info. After the session ended and they had finished talking in the room, I went up and introduced myself and mentioned that I had an internal project name and I was wondering if they knew anything about it. Once I told them the code name, they said let's go over here and talk. It was at this point where I finally found the right person to talk to, and I got the technical information I was looking for and got some influence over the final decision. I would like to give this VP some kudos for finally helping me out, it only took an English person to travel halfway around the world and for me to travel halfway across the country to Dallas. To meet and happen to run into each other. A fun anecdotal thing is while we were talking, they mentioned they all the employees were talking about the guy wearing a disclosure shirt at the conference. For those not familiar with Mimecast there are maybe 20 of these shirts given out and incredibly rare, it's a swag item for disclosing a vulnerability. My employer happens to have 2 people who have them.
- 12/16/2019 – Mimecast acknowledges my request to place my name on their Security Researcher Hall of fame (https://www.mimecast.com/responsible-disclosure/). It's at this point that I announce (https://www.linkedin.com/posts/wesleykirkland_responsible-disclosure-activity-6612351483000209408-JRo2) what has taken place on my LinkedIn (https://www.linkedin.com/in/wesleykirkland/).
- 12/17/2019 – My Mimecast account team reached out to me with an urgent request to schedule an emergency meeting with their CISO and product. What's been interesting to me is I know their CISO has known my name for quite a few months, due to emails sent to me. As well as one of the people from the product on this call I've been trying to get a meeting with for other issues since the beginning of November.
- 12/19/2019 – The call happens and Mimecast is now involved from a technical reviewer in this blog post.
- 12/20/2019 – I notice that the fix has been rolled out worldwide, while testing I discover another vulnerability and disclose it.
- 1/9/2020 – I spoke with my Mimecast account manager and she said Mimecasts' legal and Marketing teams had no comment on any details and I'm free to post it. I would like to say I mentioned the other vulnerability and they did not have comments on it happening before this post was released.

Reflections

What could I have done better?

Since I had never done this before I disclosed the vulnerability through my employer via their support channel instead of independently. If I would have disclosed independently I would have been better protected and removed my employer from the risk. I could have also observed the security communities 90 days responsible disclosure period which I was unable to follow. This breaks down to morals for me, as well as my employer was personally affected by this vulnerability as well as every Mimecast customer in the world. I should have done more Googling and found their disclosure submission system (https://www.mimecast.com/responsible-disclosure/).

What I did right.

No matter what faults happened from Mimecast or me from not disclosing it properly. The internet and Mimecast's customers are now safer. I'm proud of myself for being able to work with such an established company and to be able to find something of this scale. I just wish Mimecast would have appropriately communicated due to the unique situation I was in.

Things I could have done

I could have easily expanded this to more sophisticated attacks though, that would dive deeper into social engineering which this vulnerability was not about.

BIMI – A very new protocol called BIMI (Brand Indicators for Message Identification) requires your domains DMARC policy to be on 100% quarantine or reject status. At this point, you're able to place an SVG image that will be displayed right alongside your authenticated email. Since we were able to pass DMARC if a brand fell into the BIMI beta program and I was able to spoof them and pass DMARC via the vulnerability. It would have been possible to show the company's logo alongside our spoofed email. BIMI is so small right now I'm not sure if it would have possible to provide a real-world POC, though the concept is valid.

Mimecast (https://wesleyk.me/tag/mimecast/) /
Security (https://wesleyk.me/tag/security/)
Wesley Kirkland
January 10, 2020January 13, 2020

Email /
Mimecast /
Security /
Vulnerability

2 thoughts on “My first vulnerability – Mimecast Sender Address verification”

Allen Ferdinand says:
January 16, 2020 at 2:32 pm
Seems that DMARC wouldn’t have passed if the company’s policy was only DKIM, or if they used a firewall vendor that doesn’t use spf includes that include all of their clients, which seems to have been your point.

Reply

Wesley Kirkland says:
January 17, 2020 at 10:14 am
Allen,

You wouldn’t be able to do DKIM only as DMARC relies upon SPF or DKIM. The default record is going to specify the aspf tag as r (relaxed) which is allowing this vulnerability to pass through. As for a firewall issue, this could happen if you use EOP or something similar to say our mail only comes from these IPs. Though since it’s coming from Mimecast it’s already an authorized IP with the default Anti Spoofing entries in Mimecast.

From Dmarcian
“Specifies “Alignment Mode” for SPF. Authorized values: “r”, “s”. “r”, or “Relaxed Mode” allows SPF Authenticated domains that share a common Organizational Domain with an email’s “header-From:” domain to pass the DMARC check. “s”, or “Strict Mode” requires exact matching between the SPF domain and an email’s “header-From:” domain.”

Reply