Credential Harvesting at Scale Without Malware

unit42.paloaltonetworks.com/credential-harvesting/

Brady Stout September 30, 2021

By Brady Stout

September 30, 2021 at 3:00 PM

Category: Ransomware, Unit 42

Tags: Business Email Compromise, Cybercrime

This post is also available in: 日本語 (Japanese)

Executive Summary

While ransomware and ransomware-as-a-service (RaaS) attacks have dominated much of the cybersecurity community's discussions over the past several months, criminals and hackers continue to compromise corporate, business and personal emails for financial gain. These scams, business email compromise (BEC) and personal email account compromise (EAC), continue to be the most pervasive and costly reported cyberthreats to users daily. In its latest annual report, the Federal Bureau of Investigation (FBI) identified that BEC and EAC accounted for at least $1.86 billion in losses within the U.S. in 2020, a 5% increase over losses reported in 2019. BEC and EAC accounted for 45% of all 2020 reported cybercrime losses in the U.S., and individuals over 60 years of age accounted for 11% of the reported victims.

By rough comparison, the largest known ransomware payoff to date is $40 million. The 2021 Unit 42 Ransomware Threat Report found that the average ransomware demand was $847,344 in 2020, while the average ransom paid by victims was $312,493. In the first half of 2021, the average ransom paid climbed 82% to $570,000. These figures for average ransom paid are conservative in that they only include direct monetary losses in paid ransoms. They do not include the losses associated with a company losing revenue while being forced to operate in a degraded state during an attack, and do not include resources spent investigating the breaches; they only include known attacks. Depending upon the nature of the attack and potential data breach, a company can choose not to report a ransomware attack. Ultimately, this choice makes it challenging for the cybersecurity and law enforcement communities to determine the full scope of these crimes.

One thing that all of these attacks – BEC, EAC and ransomware – have in common is that they require privileged access to targets' networks or accounts. For most actors going against targets with average-to-below-average cyber defenses, masquerading as a legitimate user or correspondent to get into a network or account remains the easiest and most cost-effective way to gain clandestine access while maintaining a low risk of discovery. As advanced persistent threats (APTs) have shown and the United States and United Kingdom governments have observed, by using legitimate credentials and publicly available techniques, malicious actors can "evade defenses and collect and exfiltrate various information in the networks." While the APTs are successfully meeting their campaign goals with brute force credential attacks, criminals, in many cases, are simply asking their unwitting victims to hand over their credentials.

Palo Alto Networks Next-Generation Firewall customers are protected with the Advanced URL Filtering security subscription and credential phishing prevention feature. In addition, Next-Generation Firewall customers are protected using the DNS Security subscription with its automatic malicious domain blocking and proactive detection capabilities.

Organizations can learn more about preventing email-based attacks such as credential harvesting with a Business Email Compromise (BEC) Readiness Assessment.

Evolving Techniques for Email Credential Harvesting

The lucrative nature of BEC/EAC scams drives criminals to continually modify and upgrade their tactics to defeat protections. One of the newer techniques integrates spear phishing, custom webpages and the complex cloud single sign-on ecosystem to trick users into unwittingly divulging their credentials. A prevalent tactic uses seemingly benign webpages that, once opened, closely mimic legitimate login screens for popular and often used services such as:

- Office 365 and Outlook (login[.]microsoftonline[.]com)
- Outlook and Hotmail (login[.]live[.]com)
- Dropbox (www[.]dropbox[.]com/login)

- Zimbra (mail[.]zimbra[.]com)

(Dropbox said in a statement, “This activity does not involve Dropbox's service. This demonstrates the increasing complexity of relying on customers to discern real from fake, and while this doesn't involve our service, we are always working with our trusted partners to be proactive and improve where and how customers are exposed to our brand and protect it accordingly.”)

When scammers use this tactic, it usually starts with a baited email enticing the recipient to open the attachment or click on the link to a webpage. The emails usually focus on some segment of business operations (including finance, human resources, logistics and general office operations) and point to an attachment or link related to topics requiring user action. These topics include remittances, invoices, outstanding payments, requests for quotes (RFQ), purchase confirmation, shipment status, voice mails or fax delivery via email, to name a few. To make the email seem more legitimate, some criminals integrate specific information about the target in meaningful ways, including within the subject of the email. Some recent email subjects include:

- OneDrive Document to {username}
- {Company Name} New FaxMail Received {DD/MM/YYYY}
- {Company Name} New FaxMail Received {DDMMYYYY}
- {username} 1 voice message received {M/D/YYYY}
- VNotes transmitted to {username}
- Mailbox Verification for {email address}

Once opened, the email presents the user with what appears to be a typical login page. In an attempt to lower suspicion, scammers often highlight the need for heightened security or that the service logged the user out. In some cases, the pages are sent with the user’s email address already included (again in an attempt to enhance the legitimacy of the request) and simply ask for the password. These misleading login screens have alerts such as:

- You need to sign in with your email to ensure you are the rightful recipient of the protected file. File is protected by [insert security vendor] for Mail Servers.
- To read the document, please enter with the valid email credentials that this file was sent to.
- Because you're accessing sensitive info, you need to verify your password.
- Authentication needed because you're accessing a sensitive document.
- This device is not recognized. For security, [company name] want [*sic*] to make sure it's really you.
- Your email account {username} has been signed out, click ok to sign in.
- Please log in to your account to view secured files
- You have been logged out! Please enter your correct Email and password!
- Get to your documents from anywhere by signing into Office.
- “Your password is security to view your fax message.

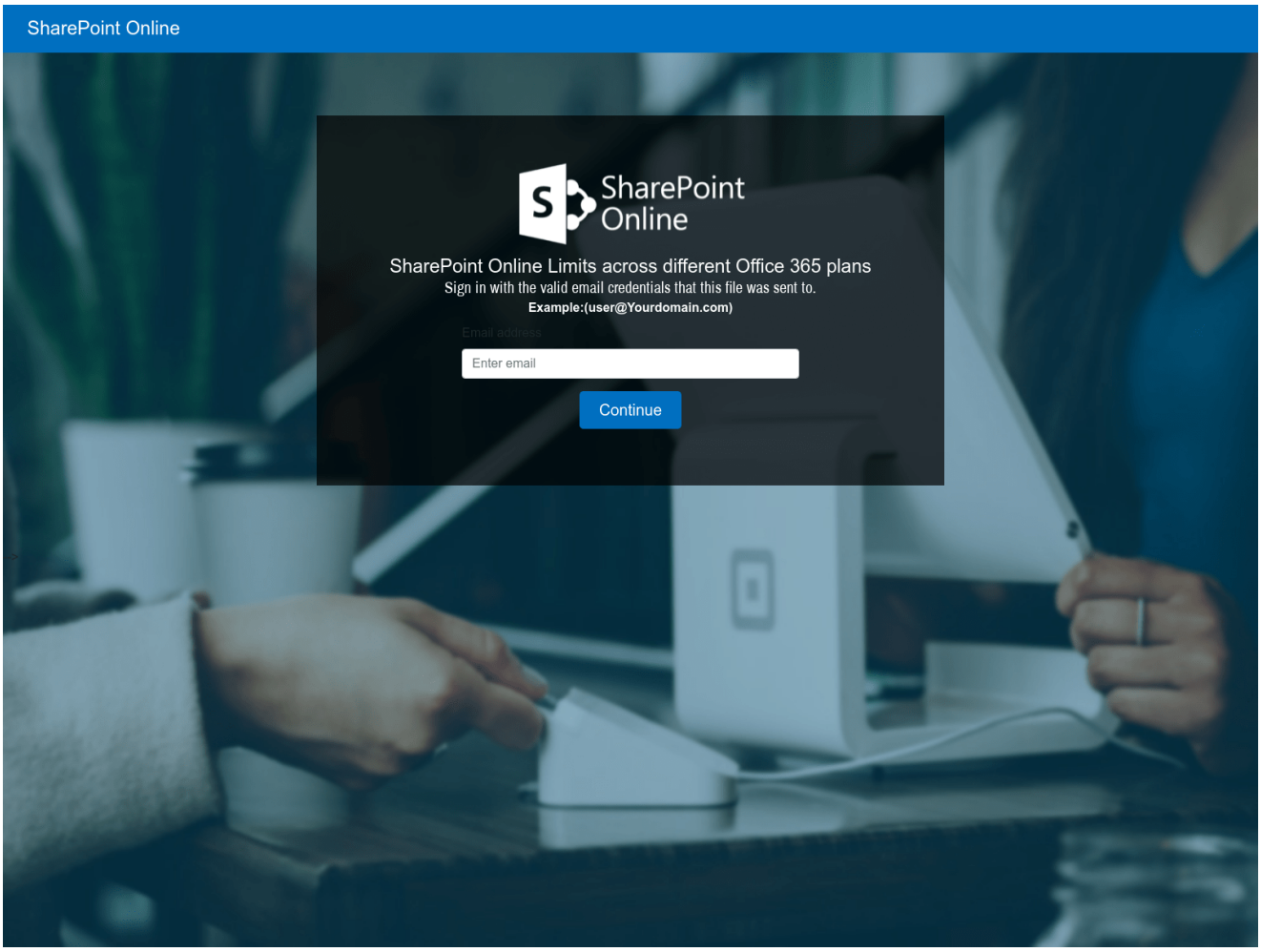

Figure 1a. Example of a malicious login request impersonating Microsoft, requiring credentials for document access.

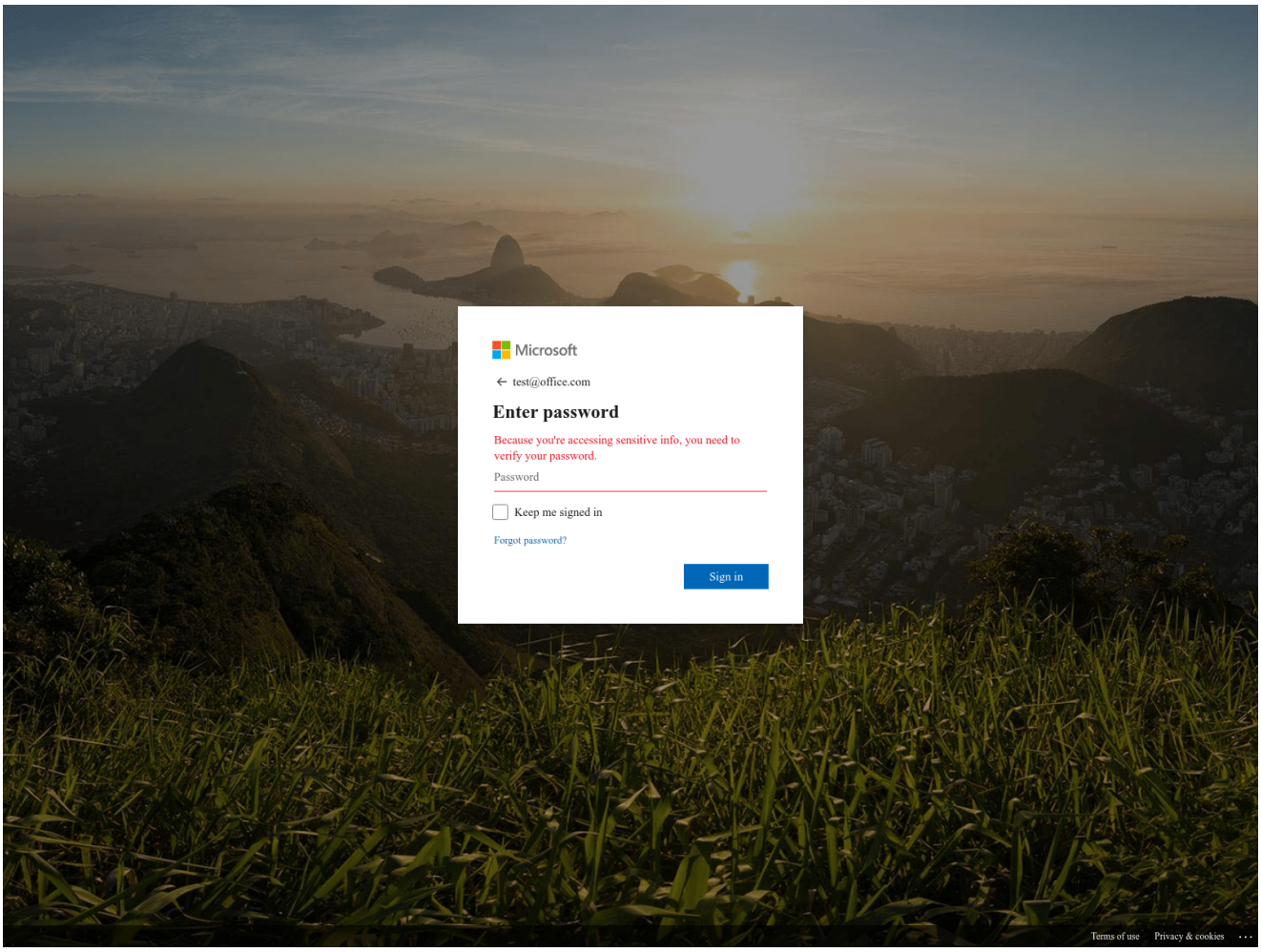

Figure 1b. Example of a malicious login request impersonating SharePoint, requiring credentials for document access.

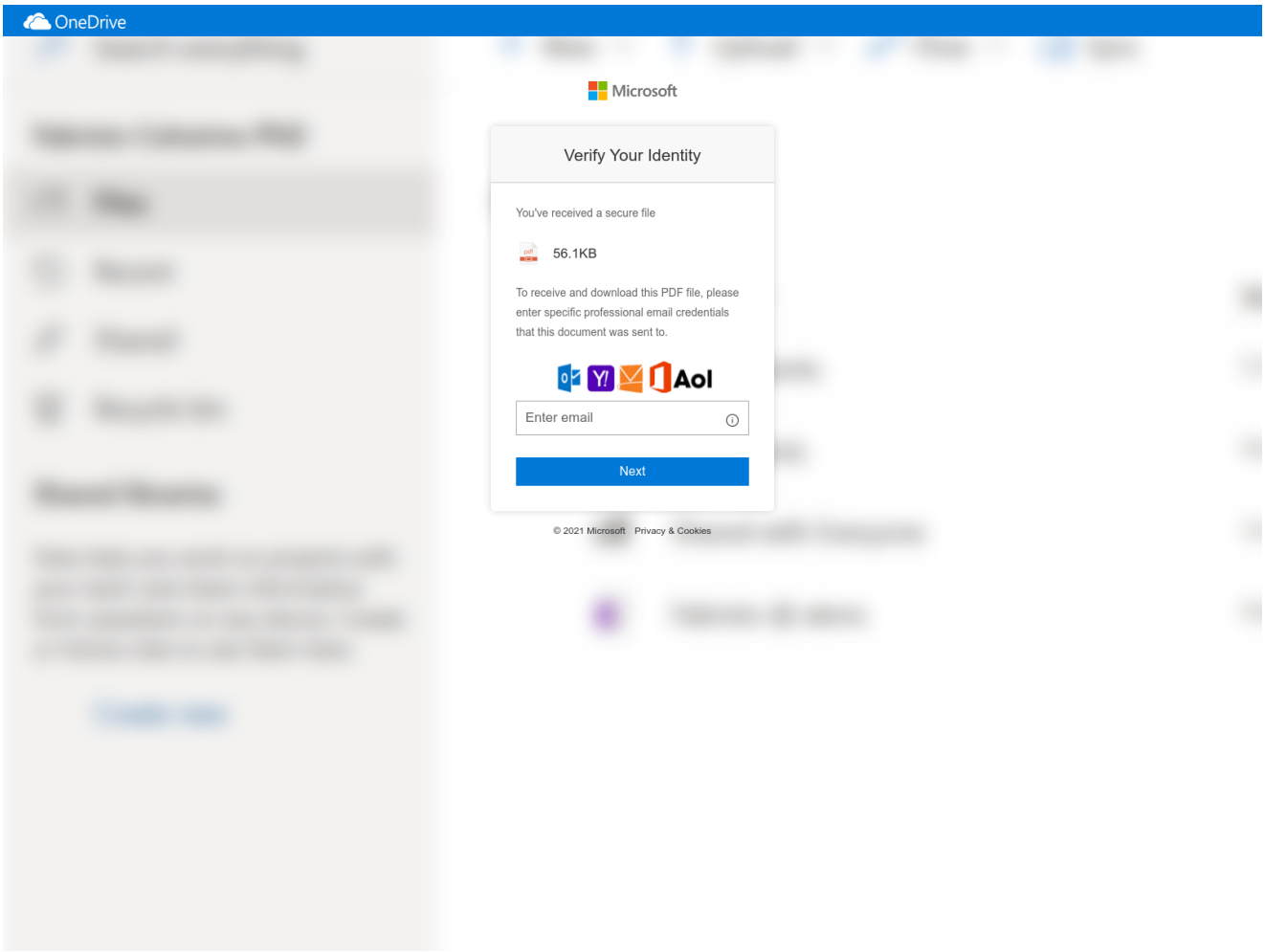

Figure 1c. Example of a malicious login request impersonating Microsoft, requiring credentials for document access.

Scammers are also adding clever tactics to further deceive users. In some instances, they are custom building their “login” templates to match the look and feel of the corporate email systems used by the specific companies they are targeting. In others, they are automatically detecting the affiliated company based on the domain portion of the user’s email address and then integrating that company’s logo into a fraudulent webpage.

```
var ind=my_email.indexOf("@");
var my_slice=my_email.substr((ind+1));
var c= my_slice.substr(0, my_slice.indexOf('.'));
var final= c.toLowerCase();
var finalu= c.toUpperCase();

$("#logoimg").attr("src", "https://logo.[redacted].com/"+my_slice);
$("#logoname").html(finalu);
$(".logoname").html(finalu);
```

Figure 2. Example of JavaScript used to identify an organization from a victim’s email address and then incorporate its logo into followup pages.

In addition, many criminals are adding logic into their code to ensure that credentials are accurately entered by the user. An incorrectly formatted email address or blank password will generate an error directing the user to retry. Some criminals are also automatically responding to the first correctly formatted attempt with "Incorrect password please try again." These techniques increase the likelihood of criminals receiving valid passwords and potentially reduces the suspicion of cautious users who perhaps first enter bogus credentials to see if the request is legitimate.

```
var my_email =email;
var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

if (!email) {
      $('#error').show();
      $('#error').html("Please enter an email address");
      ai.focus;
      return false;
  }

  if (!filter.test(my_email)) {
      $('#error').show();
      $('#error').html("That account doesn't exist. Enter a different account");
      ai.focus;
      return false;
  }
  if (!password) {
      $('#error').show();
      $('#error').html("Please enter your password.");
      ai.focus;
      return false;
  }
```

Figure 3. Example of JavaScript used to validate credentials.

Suppose scammers believe that their chances of getting a user to open a file attachment are too low or that they can create a somewhat believable fully qualified domain name. In that case, they can also simply point the user to a website on a legitimate hosting service where the above techniques are incorporated within a hosted page. Some of the recent malicious websites a user could mistakenly navigate to include:

- excel-client-login[.]azurewebsites[.]net
- excel-docs-storage[.]us-south[.]cf[.]appdomain[.]cloud
- microsoftvoicemessage-office365voicemessage-releaseandlistentov[.]s3[.]eu-de[.]cloud-object-storage[.]appdomain[.]cloud
- online-access-app[.]azurewebsites[.]net
- redirect-office365[.]web[.]app

Once a user enters and submits credentials, the web browser sends the information in an HTTP post request to a URL most often ending in *.php. As a hypertext processor, PHP enables the scammer to easily capture any received credentials, decode them and store them within a database. In addition, while the web domains enabling these scams can be purchased and maintained by criminals, we see significant use of previously compromised and coopted legitimate domains to meet these scammers' needs.

This malicious use of coopted legitimate infrastructure poses two primary challenges for network defenders. First, identifying the traffic as malicious is difficult due to it taking place between two potentially trusted networks. Second, blocking the legitimate domain, once identified as hosting malicious activities, is often not possible, as it would also block the domain's legitimate and often required content. For these reasons as well as the zero cost, hackers are increasingly relying on coopted infrastructure to meet their desired ends.

To help keep users from becoming suspicious when they fail to log in to a fake site, scammers commonly incorporate one of the following:

- Redirection to the legitimate site the user believes they are logging in to, which – if already logged in – will take them directly into their account, thereby increasing their sense of the request's legitimacy.
- A "service unavailable error" recommending they try again later.
- A "file not found" error.
- A "scanned file locked" error and "redirecting back to your account," which then redirects the user back to their legitimate inbox.
- Generic content.
- Content custom-crafted for the phishing attempt.

Once criminals have valid user credentials, they are one step closer to defrauding a company or user of their money. Using the harvested credentials, a criminal will conduct an initial reconnaissance of the user's documents, transactions and correspondence. Armed with this information, a criminal is now better informed to be able to: identify additional targets of value, understand normal business processes and approval chains, leverage the user's documents or shared file access to create custom phishing documents, and use the account for financial gain or to pivot into more lucrative environments by masquerading as the account user.

Conclusion

Malicious tactics such as those described above can be very challenging for an enterprise or user to detect. In addition, most cybersecurity products will often not automatically detect these activities as malicious due to scammers using exact copies of legitimate webpages in their scams and not incorporating trojans, spyware, keyloggers or other malware into their harvesting attempts. Unit 42 researchers recommend the following to mitigate the risk of email compromise posed by the above tactics:

- Implement multi-factor authentication for all business and personal accounts.
- Continually update and train users on the evolving tactics and social engineering used in BEC/personal EAC scams.
- Incorporate the Zero Trust mindset, "Never trust, always verify."
- Ensure users and administrators understand that even if a file successfully passes a virus scan, it could still have a malicious purpose.

- Ensure users understand which services are single sign-on and the legitimate URLs for accessing those accounts.
- Change passwords regularly, use one complex password per account and use a password manager to keep track of credentials.

Palo Alto Networks Next-Generation Firewall customers are protected with the Advanced URL Filtering security subscription and credential phishing prevention feature. In addition, Next-Generation Firewall customers are protected using the DNS Security subscription with its automatic malicious domain blocking and proactive detection capabilities.

Organizations can learn more about preventing email-based attacks such as credential harvesting with a Business Email Compromise (BEC) Readiness Assessment.

Hashes of Generic JavaScripts/HTML Used in These Techniques

Example of Basic JavaScript
e431d360ace31e17596b7017e1a11009aa3a02000d319919a266136ec11be479

Example of "+my_slice" and Re-Direction After Two Attempts
3ed082d4de4fe1d4155cb605276eecee07d11888c6cf45ffb7d9f00c6bc036b1

Example of "+my_slice" Repurposed by Actor Potentially Located in Southeast Asia
b920ac8eb9b1ac92e64fdd2df083dc7a84bca3dedcb42fd2b3959f76711607f7

Example that Prevents Shortcuts (Such as Ctrl-S) Using JavaScript KeyCodes
58ebf83741da3805955edfcf83f2b1af56a320d76954c243991fed92faeebe63

Additional Resources