URL Analysis

URLs are often the main attack vector in phishing emails, serving as gateways to malicious websites designed to steal credentials or install malware. This section covers URL structure analysis, common spoofing techniques, and various tools for safely analyzing suspicious links. Understanding how attackers manipulate domains, subdomains, and URL components is important for identifying threats before they can cause damage.

Anatomy of a URL

Likely the most important part of any malicious email. Below is a common URL structure:

Protocol: For websites, this is usually https, and http would be suspicious. However, other protocols like ftp are also sometimes used.
Subdomain: This is a segment of the main domain. For example, mail.google or docs.google
Domain: The domain of the website. This is a unique name to distinguish it from other websites in the same TLD.
TLD: Top Level Domain. Specifies what kind of domain it is. .com .org .net etc etc. Domain+TLD should be unique. As long as these are correct you can be sure the website is legitimate and not spoofed or typosquatted.
Hostname: The complete server address - the full name that identifies the specific server or computer hosting the website.
Subdirectory: Folder structure in the URL. Organizes content into logical sections on the website.
File: The specific page or resource. The actual document or script being accessed
Path: The route to the specific resource. The directory structure and filename that tells the server exactly which file to serve.
Parameter: This gives the page or resource variables that it can use. For example, timestamps for videos or search terms for a database. Comes after a question mark sign, and multiple variables are separated by ampersands.

Here is an example of a malicious URL:

http://apple-login.dnsdyn.net/ODFTY/index.php?email=asimpson@example.com

- It uses http instead of https. While this can happen with older or misconfigured websites, this would be highly unusual for a company like Apple.

- The domain is not Apple related, while the subdomain is, so this looks like sub-domain spoofing. Not everyone knows about the structure of URLs and might assume just from the sub domain that this is an Apple URL.
- The folder ODFTY is very non-specific and random. Proper websites usually have proper folder structure.
- The parameter is the recipient's email address, so the website can automatically fill it in and pretend that this was a website that the victim visited before and has their email saved.

Subdomain spoofing is often something that happens with website creators like webflow or wix

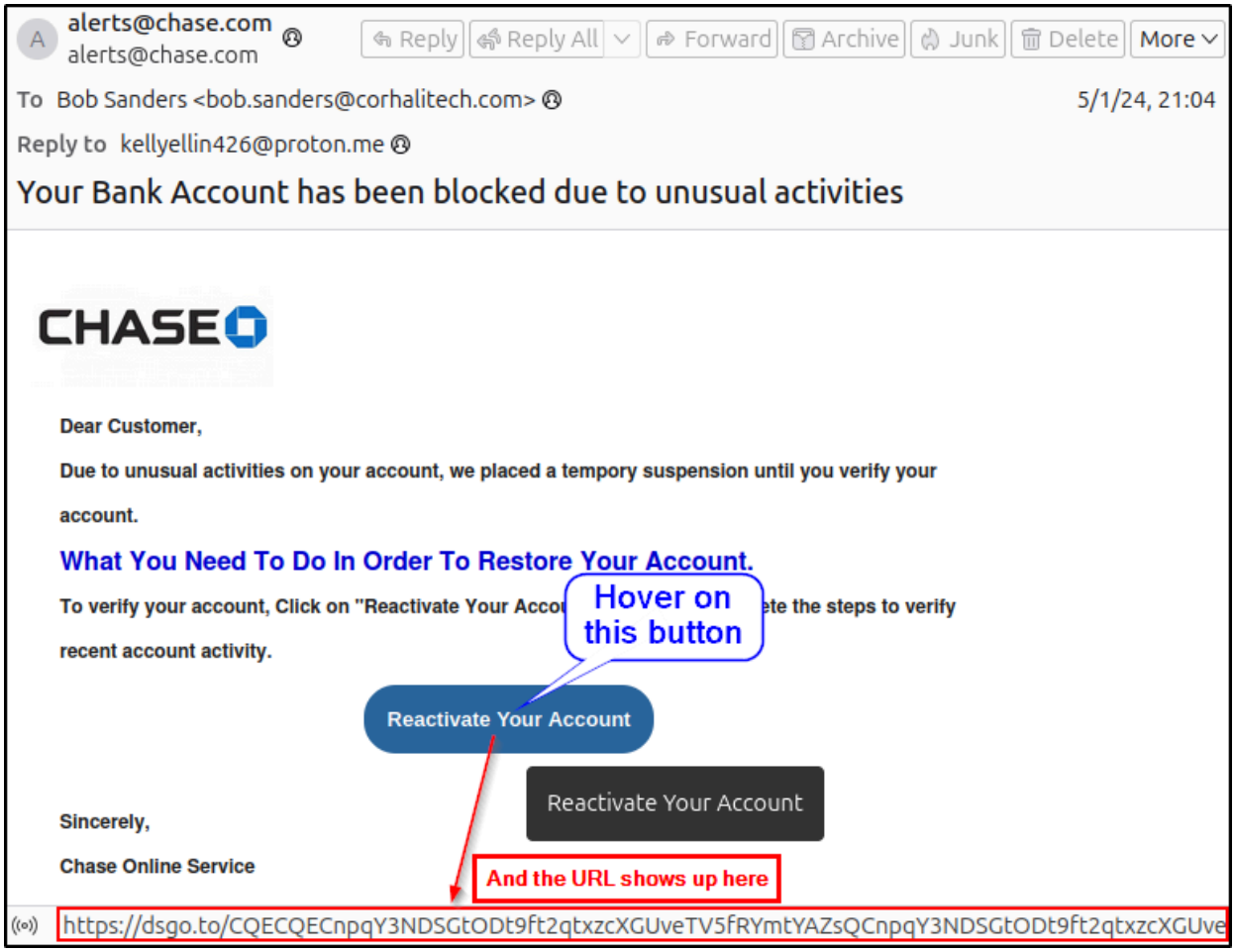
- https://metamask-restoration-support.webflow.io: metamask is a crypto platform, and would not host their site on webflow, which is a website builder
- https://revalidatenow.wixsite.com/my-site/: **wixsite.com/mysite/** is something you'll often see with domain spoofing.

Also be careful of typosquatting:

- https://www.micosoft.com/account: Microsoft is misspelled. Clear typosquatting
- https://connect.secure.wellsfargo.com/auth/login/present?origin=cob&LOB=CONS: This website uses sub-domains, but the main domain is spelled correctly and is the real Wells Fargo domain. This is an actually legitimate domain.

Practice

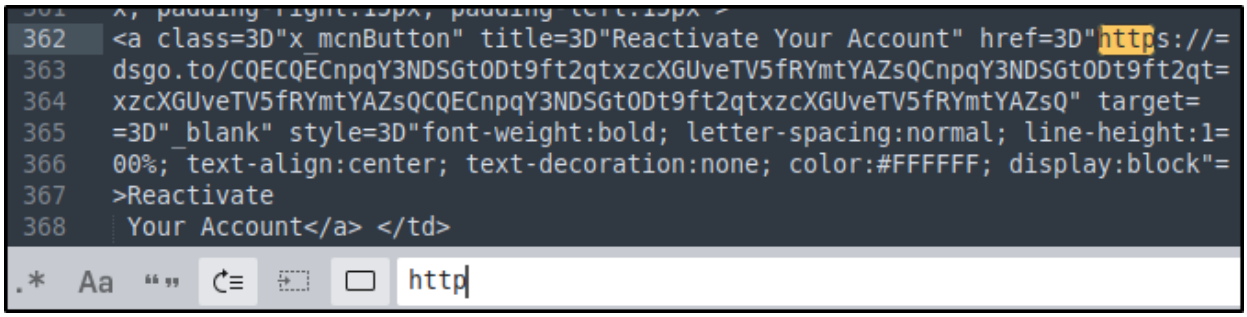
In this email example, you can check URLs even in the email client itself by hovering over the link or button:



Or you can right click the URL or button and select Copy Link Location. Note that the name of the option might be different depending on device/browser.

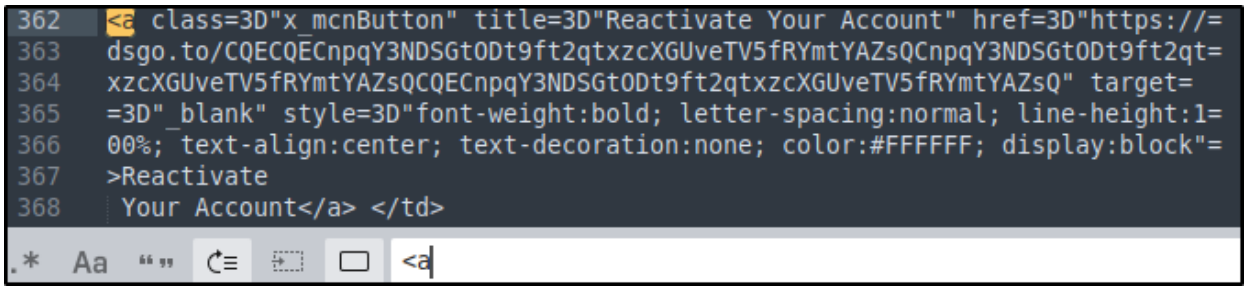
The problem with this method is that it relies on visually finding the links and buttons. You might miss links that are hidden, such as links where the color and font were changed to match the other text. Or an image might be clickable. So you would need to hover over every image and word to make sure you are not missing any links.

A better way is to save the email eml file and open it in a text editor. Then you can search for strings that appear often in URLs, such as http:



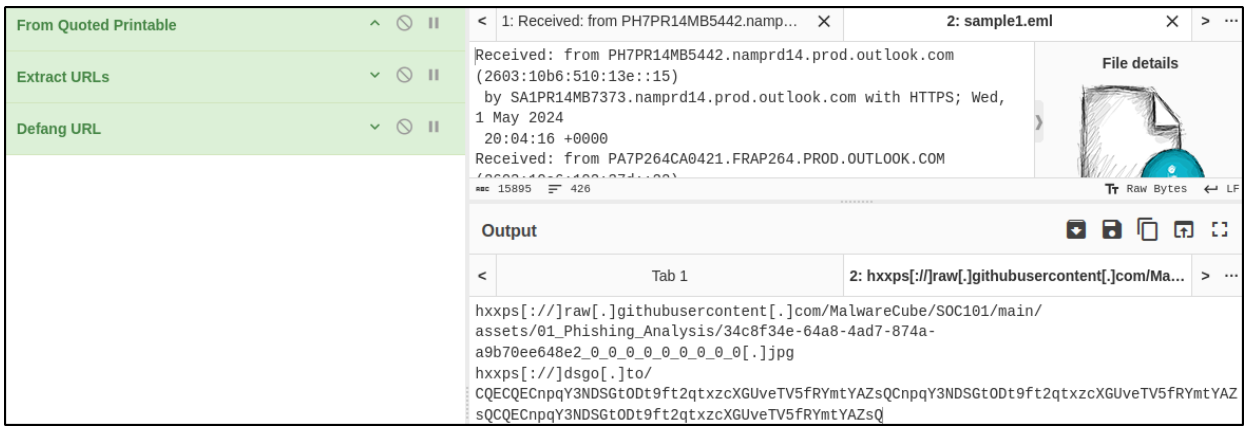
Note that not all hits will be a link, as http may be contained in headers to indicate what protocols were used.

Other ways to search for URLs is by searching for <a, which is an html anchor tag used for hyperlinks:



Not only is this method more reliable, you do not run the risk of accidentally clicking on one of these malicious links.

You can also use [CyberChef] to extract the URLs. Cyberchef even has an open file button that lets you upload the eml file directly. However, keep in mind encoding. The sample eml file was encoded using Quoted Printable. Without first decoding it, the URLs do not show up correctly. Additionally, Cyberchef has a Defang URL option that makes sure URLs are not clickable. That way they can be safely copied into reports, without running the risk of accidentally clicking them.



The URLs are defanged by changing http to hxxp, and by putting :// and periods within square brackets. So https://google.com would become hxxtp[://]google[.]com, which does not automatically convert to a hyperlink like the non-defanged URL.

Malware extraction script

The course recommends this Github repository by MalwareCube, which contains an Email IoC Extraction script. As per the repository's ReadMe: "This Python script is designed to aid in email forensic analysis by extracting various components from email files such as IP addresses, URLs, headers, and attachments." Sounds very useful.

The script can be downloaded by going to the folder where you want it, opening the folder in a terminal, and then using `wget {url}` as can be seen in this screenshot:

```
tcm@SOC101-Ubuntu:~/Desktop/01_Phishing_Analysis/Tools$ wget https://raw.githubusercontent.
com/MalwareCube/Email-IOC-Extractor/refs/heads/main/eioc.py
```

And this script can then be used by using `python3 eioc.py {file path}` as shown in this screenshot:

```
tcm@SOC101-Ubuntu:~/Desktop/01_Phishing_Analysis/Tools$ python3 eioc.py ../03_URL_Analysis/
sample1.eml
Extracted IP Addresses:
==================================
185[.]70[.]40[.]140 - Plan-les-Ouates, Geneva, CH, ISP: AS62371 Proton AG
10[.]167[.]242[.]41

Extracted URLs:
==================================
hxxps[://]raw[.]githubusercontent[.]com/MalwareCube/SOC101/main/assets/01_Phishing_Analysis
/34c8f34e-64a8-4ad7-874a-a9b70ee648e2_0_0_0_0_0_0_0_0[.]jpg
hxxps[://]dsgo[.]to/CQECQECnpqY3NDSGtODt9ft2qtxzcXGUveTV5fRYmtYAZsQCnpqY3NDSGtODt9ft2qtxzcX
GUveTV5fRYmtYAZsQCQECnpqY3NDSGtODt9ft2qtxzcXGUveTV5fRYmtYAZsQ

Extracted Headers:
```

While this is a great automation tool, it is not completely foolproof, so manual analysis may sometimes still be required.

Tools
There are various tools out there to make analysis easier. However, the Cybersecurity field keeps evolving, and new tools and new attack vectors pop up all the time. Sometimes you may have to create your own automation tools to work more efficiently.

URL reputation checking

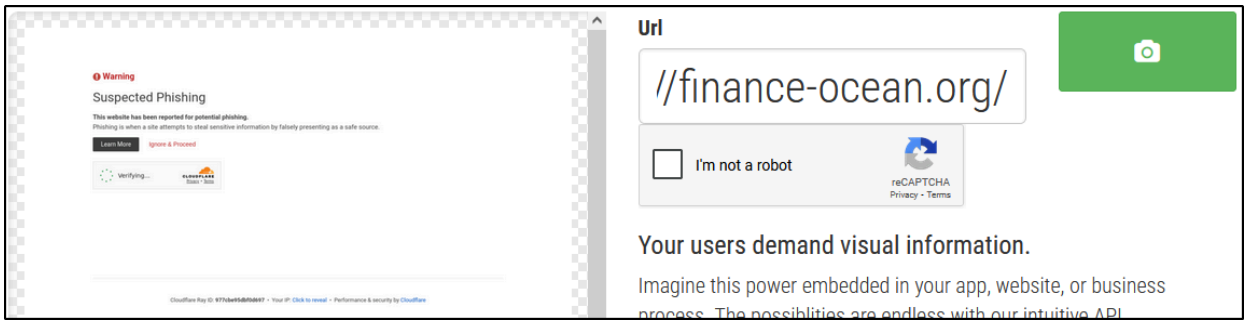
There are several tools that can be used to check the reputation of a URL, but they are also not foolproof. Just because a URL is not determined to be malicious by a tool does not mean that it is safe. It could just mean that the URL has not been submitted or reported before.

Malicious websites are often taken down quickly, either by people reporting the websites, or by the ISPs and registrars being made aware of them. Or services like Google blocking the websites. So a lot of the URLs in this course will no longer work (don't test it though, unless in a sandbox and you know what you are doing).

Phishtank and URLhaus
For actual malicious URLs, you can use PhishTank, which is a database of malicious URLs, used for research and reporting purposes. URLhaus is a similar database.

URL2PNG
To view the website without accessing it, there is URL2PNG, which shows a screenshot of the website without you having to go there. This can give you an indication of whether the website may be malicious (messy formatting, grammar mistakes, low quality logos), but is not foolproof. Some malicious websites look very benign.

An example from Phishtank I copied: hxxps[://]finance-ocean[.]org/
Which shows the following on URL2PNG:



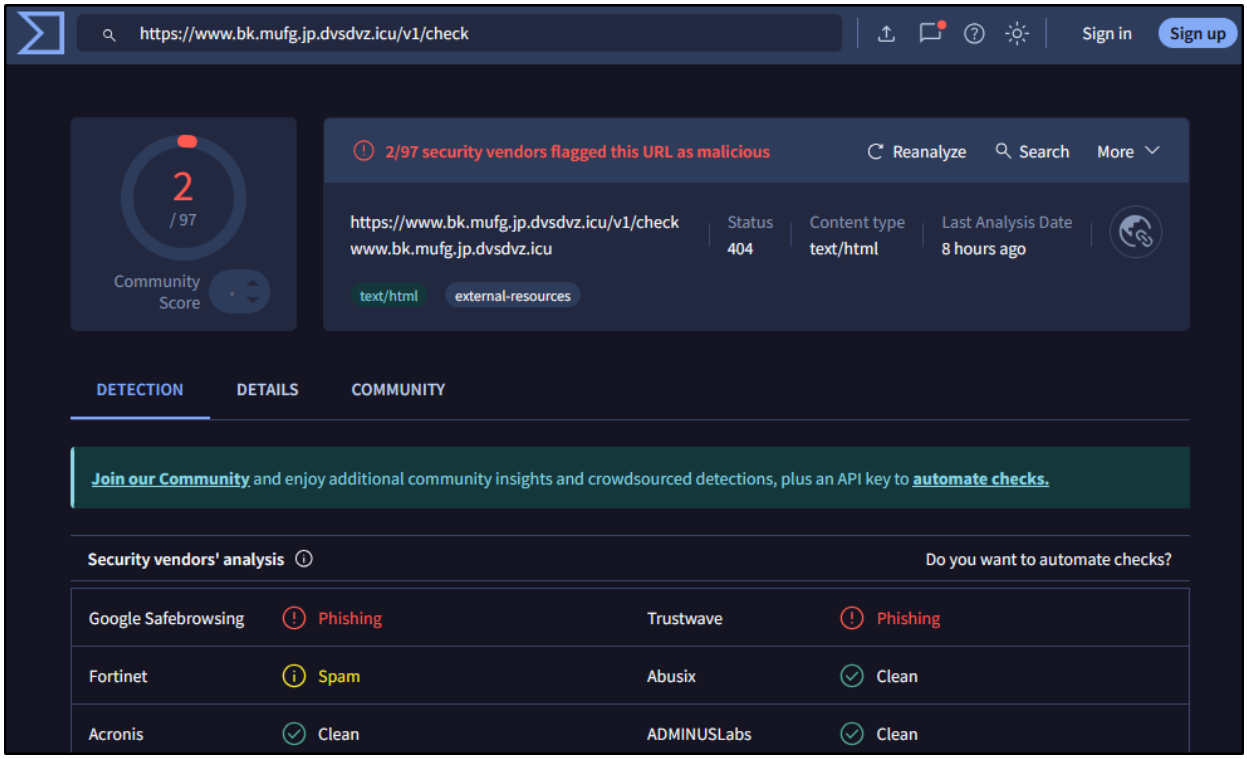
From the URL it looks like it impersonated a bank, and the screenshot shows that it was already reported and deemed suspicious by Cloudflare.

Note that I tried a number of links from PhishTank, and most of them did not produce a screenshot, so this tool may not be as useful all the time.

URLscan
A better tool may be urlscan.io. Be sure to use a private scan if you do not know whether the URL contains sensitive information. This tool lets you view the url in a sandbox environment, takes a screenshot, provides IP and reputation information, as well as providing information on when the domain was created. Very recent domains are a huge red flag.

VirusTotal

VirusTotal is the go-to tool. Aggregates multiple anti-virus engines and other security products or tools to scan files, URLs and hashes. In the below case, 2 out of 97 security services deemed the link that I copied from PhishTank as malicious:



VirusTotal also has a Details tab that shows domain history, redirect chain, IP information, response headers, and hashes.

URLvoid

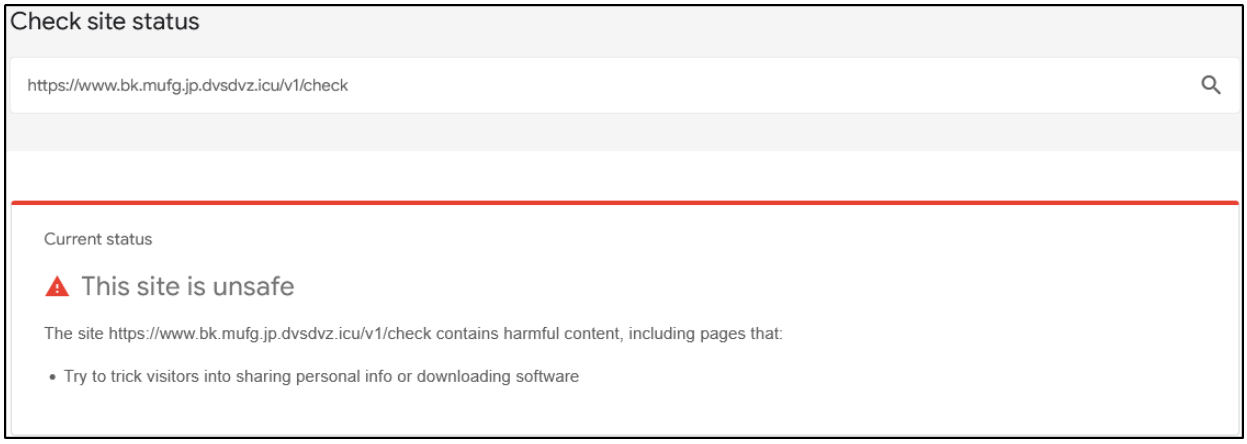
Another website scanner is URLvoid, which works similar to VirusTotal, but with different security services. If VirusTotal misses something, URLvoid might still catch it. But with very new domains, both might miss it due to the URL not having been reported yet.

Wannabrowser

To view more information on a website, you can use **Wannabrowser**. Wannabrowser shows the HTML body and Response Header of the website, which allows you to see if there are any JacaScripts, URLs or redirects.

Google's Safe Browsing

You can use Google's Safe Browsing to check whether Google deems a URL safe or not.



Base domain
When analyzing URLs, be sure to also scan the base domain. For example, the below URL has several sub domains:

hxxps[://]www[.]bk[.]mufg[.]jp[.]dvsdvz[.]icu/v1/check

But the base domain is dvsdvz[.]icu.

The full URL might have multiple sub domains, sub directories or parameters. Each of these could be their own phishing campaign. If so, scanning the base domain could help getting all of it flagged at once. Or the attacker could be using a legitimate service to conduct their malicious phishing operation. Or it is even possible for a legitimate website to have been compromised and used for phishing.

Legitimate services used for malicious purposes

Threat actors will often host their malicious files or pages on legitimate websites such as Google Docs. In this case, any scans will come out clean, as Google is a legitimate website. For example, someone could host an image of a security alert for Amazon on Google Docs, which if clicked, goes to a malicious website. However, sometimes you might get a Google Drive link, in which case you would need to download the file and perform attachment analysis, which will be covered in the next section.

Summary

URL analysis focuses on examining link structure and reputation to identify malicious destinations. Key techniques include understanding URL anatomy (protocol, subdomain, domain, TLD), recognizing spoofing methods like subdomain spoofing and typosquatting, and using safe analysis methods to avoid accidental clicks. Multiple reputation tools (VirusTotal,

URLscan, PhishTank) provide different perspectives on URL safety, but clean scans don't guarantee legitimacy. Attackers increasingly abuse legitimate services like Google Docs or website builders, making domain reputation alone insufficient for threat detection.

Red Flags Checklist

- ☐ HTTP protocol instead of HTTPS for major companies/services
- ☐ Subdomain spoofing (legitimate-sounding subdomain on unrelated domain)
- ☐ Typosquatted domains (misspelled legitimate company names)
- ☐ Website builder domains (webflow.io, wixsite.com) claiming to be major companies
- ☐ Very recent domain registration dates
- ☐ Random or nonsensical subdirectory names (like "ODFTY")
- ☐ Email address in URL parameters (pre-filling victim information)
- ☐ Multiple redirects or suspicious redirect chains
- ☐ Base domain differs from expected legitimate domain
- ☐ Legitimate services (Google Docs, OneDrive) hosting suspicious content
- ☐ URLs that don't match the context of the email sender
- ☐ Multiple security engines flagging the URL as malicious
- ☐ Suspicious IP addresses or hosting providers in URL details