WebRTC Local IP Leak Test 🍌

This script uses a pregenerated foundation lookup table to find local IP address for ICE candidates that render local mDNS hostnames seen as `xx-xxx-xxx-xx.local`.

☆ Star 15 ⊙ Issue 1

Begin test

Background

Modern browsers hide user's local IP address by returning a `[rand].local` placeholder resulting in many popular test websites such as BrowserLeaks WebRTC Leak Test showing no "Local IP Address".

Similarly creators of the (useless) stealth browsers typically mask local IP by merely swapping the address parameter in the RTC report and sometimes setting a random `foundation`. This allows anti-bot vendors to easily pinpoint the malicious visitors using these type of solutions.

A reference `libwebrtc` implementation `p2p/base/port.cc#L99` takes a local IP address along with used protocol and type and calculates a CRC32:

```
std::string Port::ComputeFoundation(const std
    const std::string& protocol,
    const std::string& relay_protocol,
    const rtc::SocketAddress& base_address) {
    rtc::StringBuilder sb;
    sb << type
        << base_address.ipaddr().ToString()
        << protocol
        << relay_protocol;
    return rtc::ToString(rtc::ComputeCrc32(sb
}
```

The test above uses a mapping of over 23'000'000 hashes for local IP ranges to corresponding parameters. In a production environment this mapping would be rather stored server-side.

Footnotes 🔐

This website does not collect, send or store your data. With a small exception for ICE candidate gathering where public STUN server from Google is used.

Source code Author