HTTP/2 & WebSockets Are Gonna Change the Pen Test World. Are You Ready?
WEBSOCKETS

RFC 6455 in 2011

- Provides full-duplex communications over a TCP connection
- Also provides bidirectional communications

If the connection starts HTTP or HTTPS the switch to WebSocket is an upgrade HTTP 101 to WS:// or WSS://

- Currently supported by most browsers and servers
- There are clients other than browsers that use WebSockets
- The application must also support it

WebSockets address limitations in HTTP/1.X and AJAX

- WebSockets either client or server can send data at any time
- May have its own port and not require initiation over HTTP in the future
### FRAME HEADERS

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| F | R | R | R | opcode | M | Payload len | Extended payload length |
| I | S | S | S | (4) | A | (7) | (16/64) |
| N | V | V | V | S | | | (if payload len == 126/127) |
| 1 | 2 | 3 | K | | | | |

- Extended payload length continued, if payload len == 127
- Masking-key, if MASK set to 1
- Masking-key (continued)
- Payload Data
- Payload Data continued ...
- Payload Data continued ...
SECURITY ISSUES

Designed for performance and convenience
Little security was built into the protocol
  • No authentication beyond upgrade request is performed
  • HTTP cookie is passed over during the handshake
  • Same Origin Policy is not enforced

Cross Site WebSocket Hijacking (CSWH) takes advantage of this
  • Attacker intercepts the upgrade request and cookie to hijack the connection
  • Application must set an origin to prevent hijacking
  • Can also encrypt with TLS via wss://
THE ATTACKERS VIEW OF WEBSOCKETS

This is a relatively new area of security research

New technologies create challenges for defenders

- Protocol use might not be properly monitored
- Defenders might not even know it is there!

Attackers can leverage WebSockets to

- attack server side
- attack client side
- attack parsers
- bypass filtering
WebSockets have been a source of interesting vulnerabilities

- Apache, Wireshark, Chrome, OpenStack, MessageSight, Firefox, Drupal, Ansible Tower, and others
- Denial of service, remote code execution, sandbox bypass, and authorization bypass
- Likely this is just the tip of the iceberg
There is a lack of tools that can do WebSocket testing
• Most automated scanners will completely miss WebSockets
• Even fewer options from commercial vendors

Three tools useful for testing applications that use WebSockets
• Burp can proxy WebSocket traffic
• OWASP ZAP can proxy and fuzz WebSocket traffic
• Chrome offers a WebSocket client and developer tools (F12)

Beyond that
• During the mapping phase look for ws:// or wss://
• Go old school and write our own test cases and script them
• Both Python and Ruby support WebSocket clients and servers
WebSocket demo
HTTP/2 & WebSockets Are Gonna Change the Pen Test World. Are You Ready?
HTTP/2

RFC 7540 in May 2015, based on SPDY
  • Quite different from HTTP/1.X in some ways
  • Same core features, much more efficient

Addresses shortcomings in HTTP/1.X
  • The header is binary and compressed
  • The basic protocol unit is a HTTP/2 frame
  • Bidirectional full-duplex over single TCP socket called a stream
  • Different frames defined, each serves a different purpose

Most implementations TLS encrypt by default
HTTP/2 may supplant WebSockets, solve some of the same issues
Why HTTP/2 was developed

- HTTP had to evolve to support modern applications
- HTTP/1.1 is inefficient, many requests for single page

HTTP/2 is intended to be much more efficient and secure

- HPACK compresses headers with Huffman coding
- Headers only resent if changed
- Headers and data are sent in HTTP/2 frames
- HTTP/2 is bidirectional and multiplexed

Server can push data client never asked for!
Client sends a HTTP/1.1 request with an upgrade request “upgrade: h2c”

GET / HTTP/1.1
User-Agent: curl/7.41.0-DEV
Host: nghttp2.org
Accept: */*
Connection: Upgrade, HTTP2-Settings
Upgrade: h2c-14
HTTP2-Settings: AAMAAABkAAQAAP__

Server will respond HTTP/1.1 with the 101 switching protocols:
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Upgrade: h2c-14
Then the server responds with a settings frame:

```
00 00 0c 04 00 00 00 00 00 03 00 00 00 64 00 .................d.
04 00 00 ff ff
```

And the client responds with a magic to establish connection:

```
50 52 49 20 2a 20 48 54 54 50 2f 32 2e 30 0d 0a  PRI * HTTP/2.0..
0d 0a 53 4d 0d 0a 0d 0a
```

PRISM...... It was changed from START 8 days after Snowden's leak
Now HTTP/2 streams can be established between client and server
• Servers can initiate push stream to clients using PUSH-PROMISE frame
• Clients can refuse with RST_STREAM frame or even GO_AWAY frame

From: https://tools.ietf.org/html/rfc7540
This is a relatively new area of security research
New technologies create challenges for defenders
  • Protocol use might not be properly monitored
  • Defenders might not even know it is there!
Attackers can leverage HTTP/2 to
  • attack server side
  • attack client side
  • attack parsers
  • bypass filtering
Discovered by a team at Yahoo! that performed fuzzing

- CVE-2015-7219 Firefox DoS
- CVE-2015-7218 Firefox DoS
- CVE-2015-5638 H2O directory traversal
- No-CVE. 4 bugs in node-http2 discovered by fuzzing
- CVE-2015-3249 Apache Traffic Server possible remote code execution
- CVE-2015-0799 Firefox MiTM X.509 validation bypass
- CVE-2014-1582 Firefox MiTM Public Key Pinning allows spoofing

Very likely just the tip of the iceberg...
We prevent the upgrade, and test over HTTP/1.1
There are no few proxies available that support HTTP/2
  • Mitmproxy
  • Charles Proxy

No few commercial tools that support HTTP/2 currently
Some tools do exist that can speak HTTP/2:
  • Curl
  • Nghostp
  • Python Hyper and NetHTTP2 in Ruby
  • Mitmproxy
  • Wireshark
HTTP/2 DEMO!

HTTP/2 demo
adriendb@gmail.com
@adriendb
613 797-3912
GWAPT, GXPN, GPEN, GCIH,
GCIA, GSEC, CISSP, OPST, OPSA
SANS Instructor 504, 542, 560, 642
SANS Internet Storm Center Handler
Okinawan Goju Ryu and other martial arts
COURSE RESOURCES AND CONTACT INFORMATION

AUTHOR CONTACT
Justin Searle
justin@meeas.com
Adrien de Beaupré
adriendb@gmail.com

SANS INSTITUTE
8120 Woodmont Ave., Suite 310
Bethesda, MD 20814
301.654.SANS(7267)

PEN TESTING RESOURCES
pen-testing.sans.org
Twitter: @SANSPenTest

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org