Creating browser extension to hunt for low hanging fruits

BY REWANTH COOL

SECURITY CONSULTANT

About

- Security Consultant
- Open source contributor
  - Contributed to Nmap and added 17,000+ lines of code
- Full stack developer
- About Payatu
  - A boutique security testing company specializing in IoT, Mobile, Cloud – https://payatu.com
  - Products
    - Expliot – IoT Security Testing framework - https://bitbucket.org/aseemjakhar/expliot_framework
    - Cloudfuzz – Countinous Fuzzing framework
    - Hacksys Extreme Vulnerable Driver - http://www.payatu.com/hacksys-extreme-vulnerable-driver/
    - Damn Insecure and Vulnerable App for Android - http://www.payatu.com/damn-insecure-and-vulnerable-app/
  - In-house Fuzz testing Infrastructure
  - Mobile/Windows kernel/IoT exploitation training – Blackhat, Brucon, Hack In Paris, HITB and Corporate trainings

Payatu

Agenda

- Need for creating custom tools for bug hunting
- Analyze low hanging fruits for headers - CORS Misconfiguration, Host Header Injection and Clickjacking
- Creating a firefox extension to hunt for low hanging bugs

Need to create custom tools for bug hunting

- To earn more bounties/money
- Automate boring repetitive important checks on multiple endpoints with less false positives
- Comparison graph between automated testing, manual testing and testing with custom tools

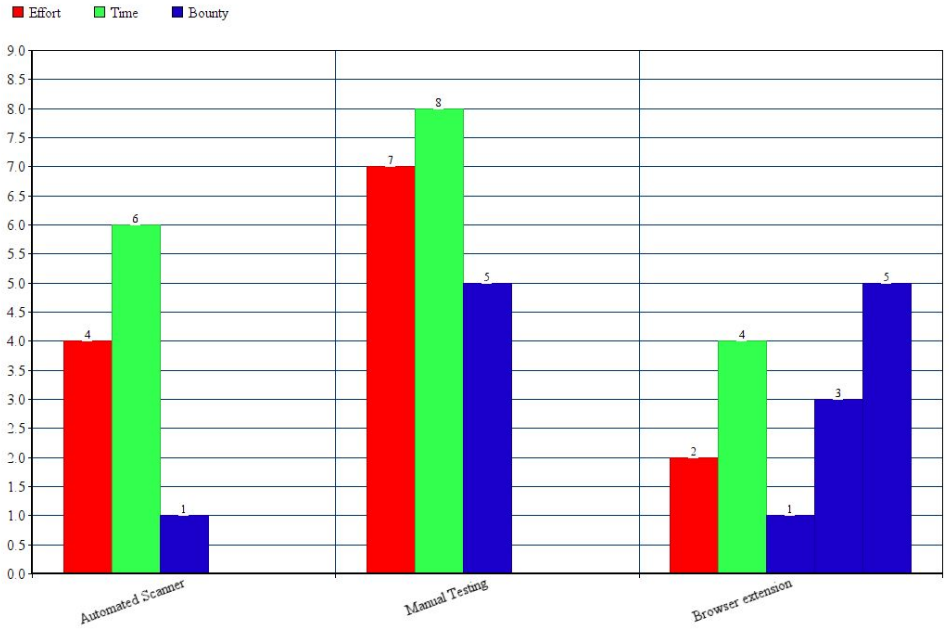
Comparison Sheet

Automated VS Manual VS Custom scans

EFFORT

TIME

BOUNTY



Analyze low hanging fruits

- CORS Misconfiguration
- Host Header Injection
- Clickjacking

CORS Misconfiguration

- Gives permissions to load scripts/request resources from other pages/domains.
- Effects -
  - If the website allows loading scripts, then attackers might be able to exploit it.
  - Allows to access restricted resources from other domains.
  - CORS bypasses the Same-origin policy which would block a AJAX requests from accessing data on a web page unless it is coming from the same origin.
- https://hackerone.com/reports/235200 ($1000)
- $(curl -I URL -H “Origin: evil.com”)
- We will study CORS misconfiguration with some animations.

Host Header Injection

- The host header tells the web server which virtual host to use (if set up).
- Effects -
  - Causes redirection
  - Password Reset Poisoning(victim receives reset link as http(s)://evil.com/token/<lja830ru28f>)
- https://hackerone.com/reports/317476 ($7560)
- Modify/Add X-Forwarded-Host header and page redirects to attacker's domain

Clickjacking

- Clickjacking is used to load/embed particular iframes in a website.
- Effects -
  - Attackers can load embedded hidden iframes if options are not set properly.
- https://medium.com/@raushanraj_65039/google-clickjacking-6a04132b918a ($12600)
- If x-frame-options is missing, then the endpoint is likely to be flagged.

Payatu

Creating a firefox extension to hunt for low hanging bugs

- manifest.json structure
- browser.webRequest.onBeforeSendHeaders.addListener(...)
- browser.webRequest.onHeadersReceived.addListener(...)

Creating a firefox extension to hunt for low hanging bugs - manifest.json file

```
{
  "description": "This extension leverages headers to check for vulnerabilities.",
  /*...*/
  "manifest_version": 2,
  "name": "vuln-headers-extension",
  "permissions": [
    "tabs",
    "webRequest",
    "<all_urls>",
    "webRequestBlocking"
  ],
  // CSP *must* be defined to trigger event handler functions (required for search queries)
  "content_security_policy": "default-src *; script-src 'self'; object-src 'none'; style-src 'self' 'unsafe-
inline'",
  "background": {
    "scripts": ["js/background.js"]
  },
  /*...*/
  "version": "1.0"
}
```

Creating a firefox extension to hunt for low hanging bugs - browser.webRequest.onBeforeSendHeaders.addListener

```
// Callbacks a function to modify the original headers
browser.webRequest.onBeforeSendHeaders.addListener(
  addCustomHeaders, // Function
  {
    urls: ["<all_urls>"] // Filters/objects
  },
  ["blocking", "requestHeaders"] // Extra specs/permissions
);
```

Creating a firefox extension to hunt for low hanging bugs - browser.webRequest.onHeadersReceived.addListener

```
// Callbacks verifyHeaders function to read the response headers
browser.webRequest.onHeadersReceived.addListener(
  verifyHeaders, {
    urls: ["<all_urls>"]
  },
  ["blocking", "responseHeaders"]
);
```

Creating a firefox extension to hunt for low hanging bugs - Combining both of the above functions

```
// Callbacks a function to modify the original headers
browser.webRequest.onBeforeSendHeaders.addListener(
  addCustomHeaders, // Function
  {
    urls: ["<all_urls>"] // Filters/objects
  },
  ["blocking", "requestHeaders"] // Extra specs/permissions
);

// Callbacks a function to read the response headers
browser.webRequest.onHeadersReceived.addListener(
  verifyHeaders, {
    urls: ["<all_urls>"]
  },
  ["blocking", "responseHeaders"]
);
```

Creating a firefox extension to hunt for low hanging bugs - Layout of functions used in previously shown APIs

```
function addCustomHeaders(e) {
  /*...*/
  e.requestHeaders.push({
    "name": "Origin",
    "value": corsMisconfigurationCheckUrl
  });
  /*...*/
}

function verifyHeaders(e) {
  /*...*/
  for (var header of e.responseHeaders) {
    // Checks for injected URL in response headers
    if (header.value.match(corsMisconfigurationCheckPattern)) {
      console.log("Vulnerable endpoint found", e.url);
    }
  }
  /*...*/
}
```

References

- Ref 1 - https://github.com/rewanth1997/vuln-headers-extension
- Ref 2 - https://medium.com/@rewanthcool/firefox-vuln-headers-extension-e848b6d80d14

Payatu

Thanks!

- Q & A
- Reach me at rewanth@payatu.com
- Twitter - @Rewanth_Cool