

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

ĐỒ ÁN TỐT NGHIỆP

Đề tài:

NGHIÊN CỨU VÀ XÂY DỰNG MÃ KHAI THÁC LỖ HỒNG
PROTOTYPE POLLUTION

Giảng viên hướng dẫn: VŨ VIỆT DŨNG

Sinh viên thực hiện: CAO XUÂN SANG

MSSV: DTC175524802990008

Chuyên ngành: An Toàn Thông Tin

Môn học: Đồ Án Tốt Nghiệp

Thái Nguyên, tháng 1 năm 2022

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên: **Cao Xuân Sang**

MSSV: **DTC175524802990008**

Chuyên ngành: **AN TOÀN THÔNG TIN**

Lớp: **ATTT-K16A**

Email: **sang999hb@gmail.com**

SĐT: **0334760905**

Tên đề tài: **Nghiên cứu và xây dựng mã khai thác lỗ hổng Prototype pollution**

Giảng viên hướng dẫn: **Vũ Việt Dũng**

Thời gian thực hiện: **25/10/2021** đến ngày: **21/01/2022**

Nhiệm vụ/nội dung:

Chương I: Tổng quan về an ninh mạng, giới thiệu về JavaScript và Node.js

Chương II: Lỗ hổng Prototype pollution

Chương III: Thực nghiệm khai thác lỗ hổng Prototype pollution

TRƯỞNG BỘ MÔN

(Ký và ghi rõ họ tên)

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên em xin cảm ơn thầy Vũ Việt Dũng đã giúp đỡ và hướng dẫn nhiệt tình cho em trong suốt thời gian em làm báo cáo này.

Tiếp theo em xin cảm ơn quý thầy cô trong trường ĐH CNTT&TT Thái Nguyên đã giảng dạy và truyền cảm hứng đến tất cả những sinh viên như em, để chúng em có thêm nhiều kiến thức, mở mang thêm cách tư duy, sáng tạo trong cuộc sống.

Với những kiến thức còn thiếu sót, những kỹ năng còn hạn hẹp, em không thể tránh được những sai lầm, em mong thầy cô có thể thông cảm và bỏ qua cho em.

Xin cảm ơn bạn bè, những người đã cùng em sánh bước, đã đóng góp, động viên em những lúc gặp khó khăn, tạo điều kiện giúp em hoàn thành tốt đề tài này.

Lời cuối cùng em xin gửi lời cảm ơn sâu sắc nhất và chúc thầy cô luôn dồi dào sức khỏe, tiếp tục giảng dạy hết tâm huyết của mình cho những lứa học trò sau này để đất nước ta ngày càng có nhiều nhân tài, những người giỏi trong các doanh nghiệp, xây dựng đất nước phát triển hơn nữa.

Thái Nguyên, Tháng 01 năm 2022

Sinh viên

Cao Xuân Sang

MỤC LỤC

LỜI CẢM ƠN	2
MỤC LỤC.....	3
DANH MỤC HÌNH ẢNH	5
LỜI MỞ ĐẦU	8
CHƯƠNG I: TỔNG QUAN VỀ AN NINH MẠNG, GIỚI THIỆU VỀ JAVASCRIPT VÀ NODE.JS	10
1.1. Vấn đề với an ninh mạng hiện nay.....	10
1.2. Giới thiệu về JavaScript	12
1.2.1 Lịch sử của JavaScript.....	12
1.2.2 Object trong JavaScript	14
1.2.3 Prototype trong JavaScript	16
1.3. Giới thiệu về Node.js.....	18
CHƯƠNG II: LỖ HỔNG PROTOTYPE POLLUTION	21
2.1 Giới thiệu về lỗ hổng Prototype pollution.....	21
2.2 Tác động của Prototype pollution	23
2.2.1 Property Injection	23
2.2.2 Denial of service (DoS).....	27
2.2.3 Remote code execution (RCE)	31
2.3 Ba lỗi chính trong khi viết code gây ra lỗ hổng Prototype pollution.....	34
2.3.1 Thực hiện việc ghép đệ quy các đối tượng không an toàn (Unsafe object recursive merge).....	34
2.3.2 Định nghĩa thuộc tính qua đường dẫn (Property definition by path)	34
2.3.3 Sử dụng thư viện có chứa lỗ hổng Prototype pollution.....	35
2.4 Phòng chống lỗ hổng Prototype pollution.....	35
2.5 Lỗ hổng Prototype pollution trong thực tế - CVE-2019-7609.....	39
CHƯƠNG III: THỰC NGHIỆM KHAI THÁC LỖ HỔNG PROTOTYPE POLLUTION	45
3.1 Mô hình triển khai	45

3.2. Phân tích website chứa lỗ hổng bảo mật và viết mã khai thác	46
3.3 Ngăn chặn lỗ hổng Prototype pollution trong bài lab	58
KẾT LUẬN	60
TÀI LIỆU THAM KHẢO.....	61

DANH MỤC HÌNH ẢNH

Hình 1.1. Tạo đối tượng trong javascript.....	14
Hình 1.2. Lấy giá trị thuộc tính của object.....	15
Hình 1.3. Thêm thuộc tính cho object.....	15
Hình 1.4. Thêm phương thức cho object	15
Hình 1.5. Xóa thuộc tính hoặc phương thức của một object	16
Hình 1.6. Kế thừa trong JavaScript.....	17
Hình 1.7. Prototype chain trong JavaScript	18
Hình 1.8. Cách hoạt động của Node.js.....	20
Hình 2.1. Các thuộc tính và phương thức mặc định khi tạo object	21
Hình 2.2. Thay đổi phương thức của các object bằng một object	22
Hình 2.4. Người dùng bình thường không thể lấy được secret key.....	24
Hình 2.5. Lấy secret key bằng thông tin đăng nhập của admin.....	24
Hình 2.6. Đoạn mã ghi log	25
Hình 2.7. Lấy secret key bằng tài khoản người dùng bình thường.....	25
Hình 2.8. Mã nguồn của hàm extend trong thư viện utils-extend.....	26
Hình 2.9. Đoạn code gộp 2 object.....	27
Hình 2.10. Chạy thử hàm extend của thư viện utils-extend.....	27
Hình 2.11. Mã nguồn chức năng “gửi mã coupon, trúng tiền”.....	28
Hình 2.12. Gửi request đến endpoint /getMoney.....	28
Hình 2.13. Request có chứa payload độc hại gây lỗi máy chủ	29
Hình 2.14. Request bị lỗi ngay cả khi không gửi payload độc hại	30
Hình 2.15. Một đoạn code nguy hiểm trong thư viện node.extend	30
Hình 2.16. Mã nguồn website	31
Hình 2.17. Chức năng lấy số ngẫu nhiên	32
Hình 2.18. Chức năng unflatten object	32
Hình 2.19. Request chứa payload độc hại.....	33
Hình 2.20. Payload độc hại được kích hoạt tạo ra thư mục có tên hacker.....	33
Hình 2.21. Hợp nhất đối tượng không an toàn.....	34

Hình 2.22. Lỗ hổng Prototype pollution trong thư viện object-path	35
Hình 2.23. Sử dụng Object.freeze()	35
Hình 2.24. Sử dụng Object.create(null)	36
Hình 2.25a. Kết quả sau khi chạy lệnh “yarn audit”	38
Hình 2.25b. Kết quả sau khi chạy lệnh “yarn audit”	39
Hình 2.26. Timelion trong Kibana	40
Hình 2.27. Lỗi trong console chạy Kibana	41
Hình 2.28. Hình ảnh trong debugger chứng minh Kibana đang cố gắng tạo process node mới.....	42
Hình 2.29. Một đoạn mã trong hàm normalizeSpawnArguments có thể gây ra lỗ hổng Prototype pollution.....	42
Hình 2.30. Nội dung file /proc/self/environ	43
Hình 2.31. File /proc/self/environ trở thành một file JavaScript hợp lệ	43
Hình 3.1. Mô hình triển khai bài lab	45
Hình 3.2. Giao diện website chứa lỗ hổng bảo mật	46
Hình 3.3. Mã nguồn website - phần quản lý người dùng.....	47
Hình 3.4. Hàm getServerSideProps	48
Hình 3.5. Đoạn mã khai thác lỗ hổng Prototype pollution gây lỗi 404	49
Hình 3.6. Trang web bị lỗi 404 ngay cả khi ở trang chủ	49
Hình 3.7. Mã khai thác dẫn đến tấn công DOS đối với website.....	50
Hình 3.8. Website bị lỗi 500 khi truy cập vào trang chủ	50
Hình 3.9. Hàm getInstance trong thư viện amphtml-validator	51
Hình 3.10. Hàm Validator trong thư viện amphtml-validator	52
Hình 3.11. Tìm kiếm nơi sử dụng thư viện amphtml-validator	53
Hình 3.12. Code sử dụng hàm getInstance của thư viện amphtml-validator.....	53
Hình 3.13. Thử nghiệm với vm sandbox	54
Hình 3.14. File JavaScript chứa payload độc hại.....	54
Hình 3.15. Mã khai thác RCE thông qua lỗ hổng Prototype pollution.....	54
Hình 3.16. Máy Ubuntu chiếm quyền điều khiển máy Windows.....	55

Hình 3.17. Hàm netcat.....	56
Hình 3.18. Chỉnh sửa lại hàm pollute_to_rce_nextjs và chuyển code vào máy Ubuntu.....	56
Hình 3.19. Mã khai thác hoạt động tốt trong máy Ubuntu	57
Hình 3.20. Hàm “pollute_to_dos” mới	58
Hình 3.21. Sử dụng thư viện mới để tránh lỗi hồng bảo mật.....	59
Hình 3.22. Website vẫn không bị lỗi khi bị tấn công bằng cách thức cũ	59

LỜI MỞ ĐẦU

Cùng với sự phát triển của công nghệ thông tin, công nghệ mạng máy tính và sự phát triển của mạng Internet ngày càng đa dạng và phong phú. Các dịch vụ, ứng dụng Web đã thâm nhập vào hầu hết các lĩnh vực trong đời sống xã hội. Các thông tin trên Internet cũng đa dạng về nội dung và hình thức, trong đó có rất nhiều thông tin cần được bảo mật cao hơn bởi tính kinh tế, tính chính xác và tính tin cậy của nó.

Bên cạnh đó, các hình thức phá hoại ứng dụng Web cũng trở nên tinh vi và phức tạp hơn. Do đó đối với mỗi hệ thống, nhiệm vụ bảo mật được đặt ra cho người quản trị mạng là hết sức quan trọng và cần thiết. Xuất phát từ những thực tế đó, bài viết sẽ đi tìm hiểu về kỹ thuật tấn công Prototype pollution và các cách phòng chống loại tấn công này.

Việc nghiên cứu kỹ thuật tấn công Prototype pollution, cách bảo mật loại tấn công này với mong muốn góp một phần sức nhỏ vào việc nghiên cứu và tìm hiểu về các vấn đề an ninh mạng giúp cho việc học tập và nghiên cứu.

Lý do chọn đề tài

Trong những năm gần đây, Việt Nam ngày càng phát triển về mặt công nghệ thông tin, đặc biệt là về ứng dụng Web. Hầu hết mọi người ai cũng từng nghe và làm việc trên ứng dụng Web. Website trở nên phổ biến và trở thành một phần quan trọng của mọi người và nhất là các doanh nghiệp, công ty. Bên cạnh lý do an toàn, bảo mật cho ứng dụng Web luôn là vấn đề nan giải cho người quản trị Website, một trong những lỗ hổng nguy hiểm và phổ biến gần đây là lỗ hổng Prototype pollution, luôn tạo ra các khó khăn đối với các lập trình viên website. Chính vì lẽ đó, bài đồ án tốt nghiệp đã đi vào tìm hiểu một kỹ thuật tấn công Prototype pollution và xây dựng mã khai thác nhằm mục đích tìm hiểu sâu hơn về kỹ thuật tấn công, phòng thủ trước loại tấn công này, và sau đó là chia sẻ kiến thức với mọi người.

Mục tiêu

Hiểu cách thức tấn công và phòng chống tấn công lỗ hổng Prototype pollution, xây dựng thành công mã khai thác lỗ hổng Prototype pollution

Phạm vi

Tìm hiểu các kỹ thuật tấn công, cách bảo mật, phòng thủ của kỹ thuật tấn công Prototype pollution một cách tổng quan nhất.

Bố cục của đề án tốt nghiệp

Nội dung của báo cáo bao gồm các phần sau:

- Chương I: Tổng quan về an ninh mạng, giới thiệu về JavaScript và Node.js
- Chương II: Lỗ hổng Prototype pollution
- Chương III: Thực nghiệm khai thác lỗ hổng Prototype pollution
- KẾT LUẬN: Trình bày các kết quả đạt được.