Special Report

웹 취약점과 해킹 매커니즘#10 SSRF(Server-Side Request Forgery)

■ 개요

SSRF(Server-Side Request Forgery, 서버 측 요청 위조)는 공격자가 서버에서 이루어지는 요청을 변조하는 웹 취약점 공격이다. 앞서 다뤘던 웹 취약점 공격인 CSRF(Client-Side Request Forgery)는 클라이언트의 권한을 이용해 악의적인 요청을 전송한 것과 달리, SSRF 는 서버의 권한을 이용해 악의적인 요청을 전송한다. 즉, 서버로부터 공격이 시작되기 때문에 외부에서 직접 접근이 불가한 내부 서버 및 시스템에 접근할 수 있으며, 내부 네트워크 내에서 악의적인 행위를 수행할 수 있다.

따라서 SSRF 공격에 성공할 경우, 외부 접근이 제한된 서버 내부의 자원에 접근해 주요 정보를 획득할 수 있고, 획득한 정보를 토대로 2 차 공격까지 이어질 수 있어 취약점이 발생하지 않도록 조치하는 것이 중요하다.

최근 SSRF 공격은 온프레미스 환경에서 클라우드 환경으로 변화하는 과정에서 그 시도 횟수가 증가하고 있다. 특히 퍼블릭 클라우드는 메타데이터 API 를 활용하고 있는 탓에 SSRF 취약점을 통해 공격 당할 경우 해당 서버의 환경설정, 크리덴셜 등 정보가 노출될 수 있어 주의가 필요하다.

SSRF 공격의 대표적인 사례로 2019 년 미국의 대형은행인 캐피탈 원(Capital One)에서 발생한 데이터 유출 사고가 있다. 퍼블릭 클라우드 서비스인 AWS 를 사용하는 캐피탈 원은 클라우드의 잘못된 설정으로 인해 외부 공격자가 내부 저장소에 있는 데이터에 접근할 수 있었다. 이로 인해 1 억 600 만 명에 달하는 고객의 이름, 주소, 연락처 등 고객 정보와 신용 점수, 결제 내역 등의 금융 정보가 유출됐다.

뿐만 아니라 2021 년 3 월 공개된 MS Exchange Server 의 취약점인 ProxyShell[1]과 2022 년 10 월 공개된 ProxyNotShell[2]에도 활용되는 등 SSRF 취약점은 접근 권한이 없는 외부에서 내부 서버 및 시스템에 접근할 수 있기 때문에 공격자들에 의해 활발히 악용되고 있다.

이번 Special Report 에서는 SSRF 의 개념과 동작원리, CSRF 와의 비교, 시나리오별 공격 예시, 보안대책과 우회 기법에 대해 소개한다.

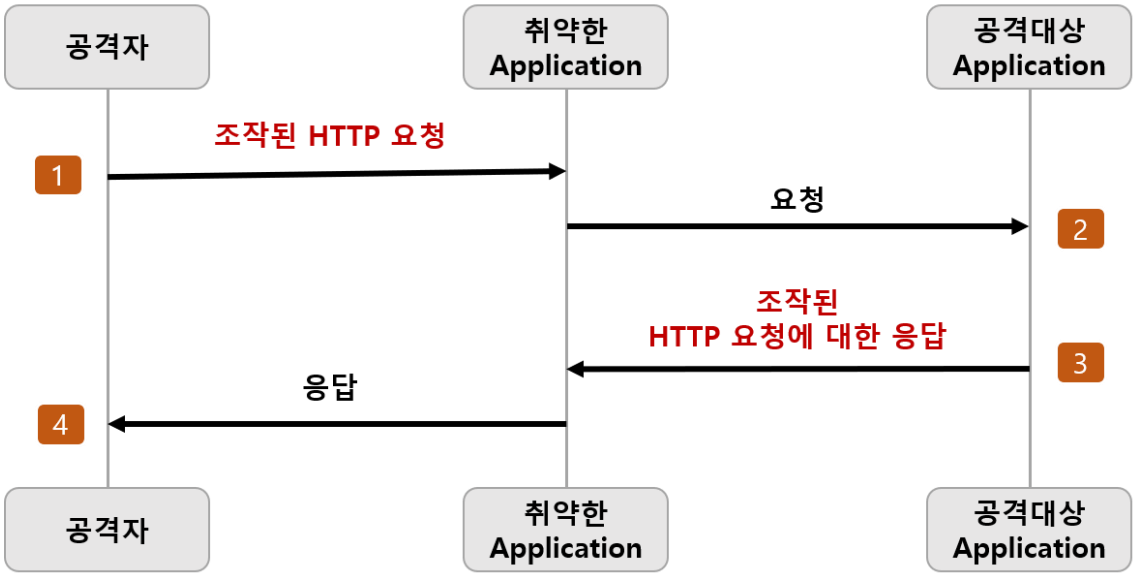
1https://www.skshieldus.com/download/files/download.do?o_fname=3.%20Research%26Technique_202104.pdf&r_fname=20220217171403287.pdf

2https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_%20Research%26Technique__202301.pdf&r_fname=20230113172426073.pdf

■ SSRF 동작원리

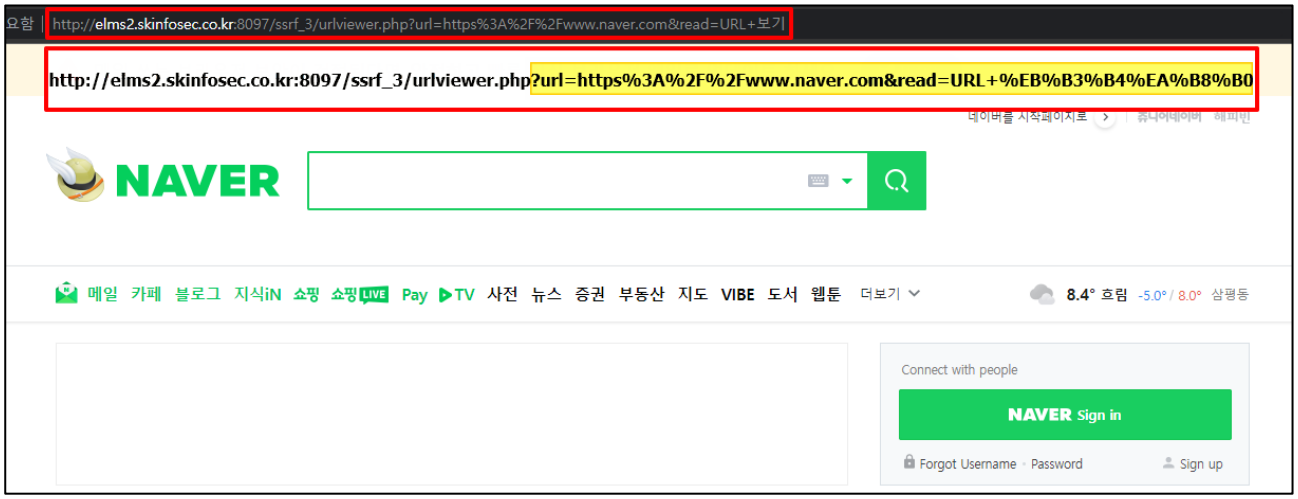
다음은 공격자에 의해 SSRF 공격이 실행되는 과정을 간단하게 나타낸 그림이다.

infosec



공격자가 SSRF 에 취약한 애플리케이션에 조작된 HTTP 요청을 전송했을 때 이에 대한 검증이 미흡할 경우, 공격대상 애플리케이션에 요청이 전달된다. 공격대상 애플리케이션은 공격자의 요청에 대한 응답을 취약한 애플리케이션에 전송하고, 취약한 애플리케이션은 해당 요청에 대한 응답을 공격자에게 전달한다. 따라서 공격자는 조작된 HTTP 요청에 대한 응답을 취약한 애플리케이션을 통해 확인할 수 있다.

SSRF 공격은 사용자의 입력값을 받아 외부 서버에 자원을 요청하는 환경에서 주로 발생한다. 이러한 환경에 대한 예시는 다음과 같다. 페이지 내 URL 뷰어를 통해 해당 웹 서버에서 URL 을 조회할 수 있다. 이때 사용자 입력값으로 받은 요청 매개변수 파라미터에 대한 검증이 미흡하거나 존재하지 않을 경우, SSRF 공격이 가능하다. 따라서 SSRF 취약점을 통해 공격에 성공하기 위해서는 외부 입력값을 받아 서버를 통해 실행하는 로직을 찾는 것이 중요하다.



■ CSRF(Cross-Site Request Forgery) vs SSRF(Server-Side Request Forgery)

앞서 1 월호에서 공격자가 타 사용자의 권한을 이용하여 자신이 의도한 동작을 서버에 요청하도록 유도하는 CSRF 에 대해 설명했다. CSRF 는 서버 내 사용자 요청에 대한 적절한 검증 절차가 없을 때 정상적인 요청과 조작된 요청을 구분하지 못해 발생하므로, 공격당한 사용자의 권한을 그대로 사용한다는 점에서 피해 범위가 달라질 수 있어 주의가 필요한 취약점이다.

CSRF 와 SSRF 의 가장 큰 차이점은 위조된 요청을 보내도록 하는 주체에 있다. CSRF 는 클라이언트 측에서 서버에 위조된 요청을 전송하도록 유도하여 공격자의 의도대로 서버가 동작하게끔 유도한다. 반면, SSRF 는 서버 측에서 위조된 요청을 보내는 것으로 웹 서버 자체를 대상으로 하는 공격이며, 주로 외부에서 접근할 수 없는 내부 시스템을 대상으로 공격을 진행한다.

CSRF 와 SSRF 에 대해 발생원인과 공격 대상, 목적, 행위에 대해 비교한 내용은 다음과 같다.

	CSRF	SSRF
발생 원인	웹 서버가 클라이언트를 신용하여 발생	사용자 입력값 검증이 미흡하여 발생
공격 대상	웹 서버	내부 서버 및 시스템
공격 목적	권한 도용, 권한 상승 등 공격자가 원하는 행위 수행	내부 서버 및 시스템 접근 후 중요 정보 유출 등
공격 행위	서버에서 제공하는 기능을 페이지에 포함시킨 후 실행 유도	외부 서버에 자원을 요청하는 서비스에 변조된 요청을 전송하여 내부 서버로 요청을 보내도록 유도

■ SSRF 공격 시나리오

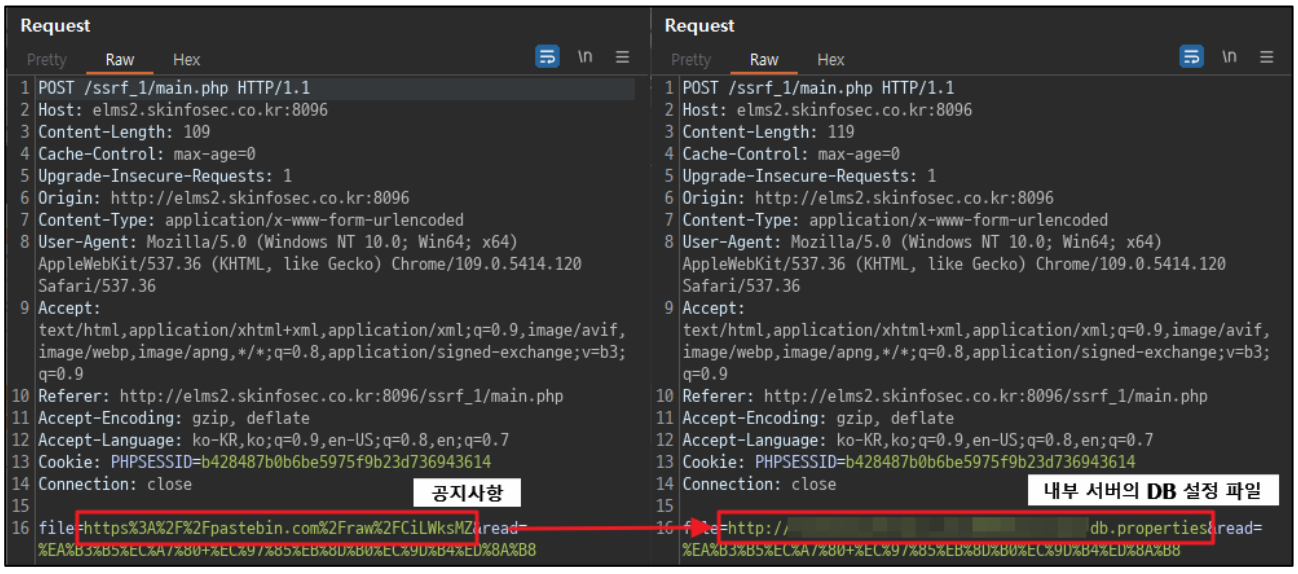
SSRF 취약점이 존재하는 환경에서의 공격 시나리오는 다음과 같다.

1) 로컬서버 파일 접근

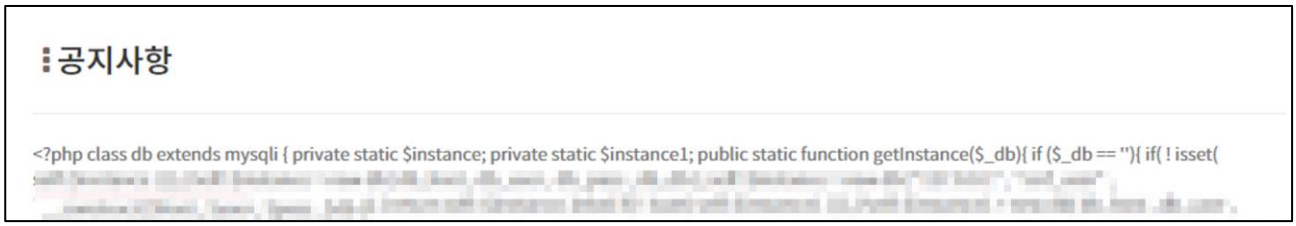
공격자는 URL Schema 를 활용하여 외부에서 접근할 수 없는 /etc/passwd/, etc/shadow, 데이터베이스 설정 파일과 같은 내부 서버의 시스템 파일에 접근할 수 있다.

step 1) 공지 업데이트 클릭 시, 내용이 보이는 페이지임을 확인한다.

step 2) 공지 내용을 반환해주는 file 의 값을 이용해 내부 서버의 DB 설정 파일을 조회한다.



step 3) 공지사항 내용 대신 내부 서버의 DB 설정 파일 내용을 반환하는 것을 확인할 수 있다.



2) 내부 웹 서버 정보 획득

외부에서 비인가자의 접근이 제한된 웹 서버에 요청을 전송하여 SSRF 공격에 성공한다면, 내부 웹 서버 자원에 접근하여 내부 정보 획득이 가능하다.

SSRF 공격을 통해 내부 웹 서버에 접근하여 내부 정보를 획득하는 공격 시나리오는 앞서 언급한 2019 년 발생한 캐피탈 원 데이터 유출 사례를 통해 확인할 수 있다.

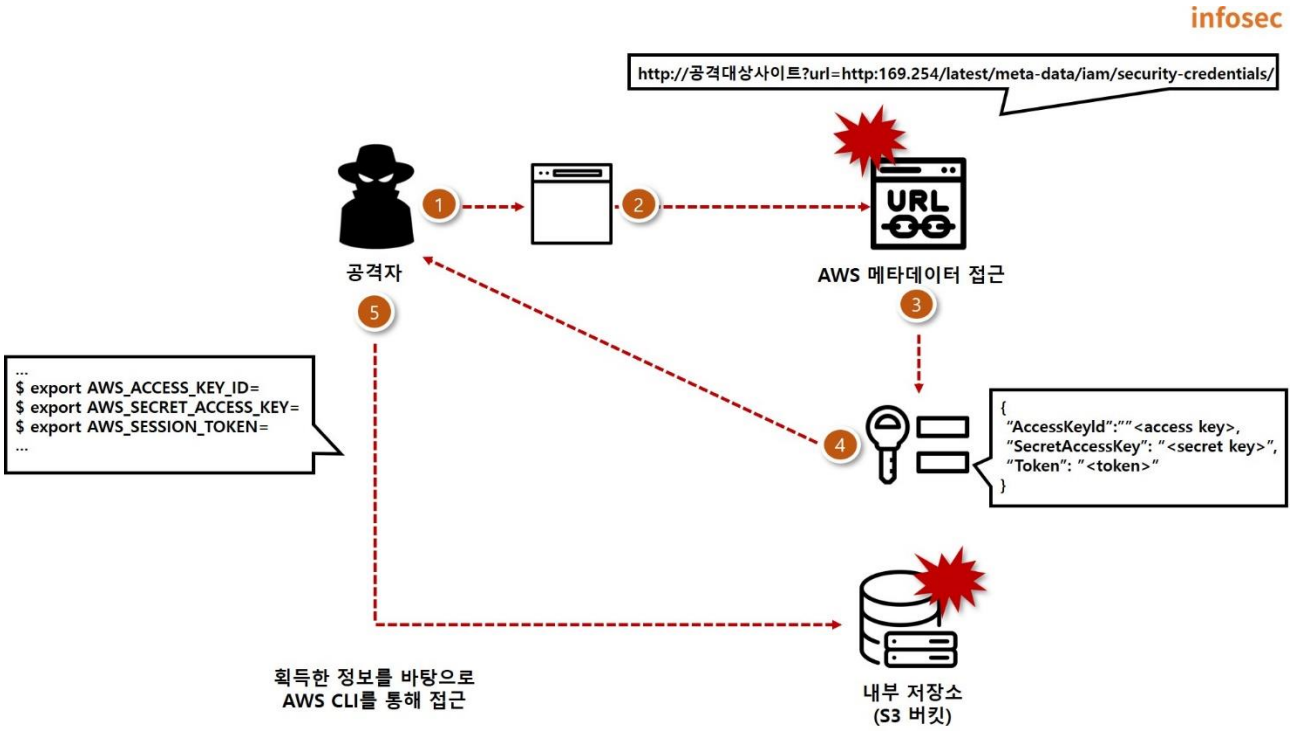
공격자는 방화벽 설정이 미흡하게 되어있는 점을 이용해 서버 측에 조작된 요청을 전송하는 SSRF 공격을 진행했다. 공격자는 AWS 클라우드로 구성된 서버의 메타데이터를 제공해주는 API 에 접근하여 IAM[3] 정보, Access Key 등 인증 정보를 탈취한 후 고객 정보가 저장된 내부 저장소인 S3[4] 스토리지에 접근했다.

해당 공격이 발생한 근본적인 이유는 사용자 측의 잘못된 역할 및 권한 설정에 있지만, SSRF 공격을 통해 내부 저장소에 접근하여 큰 피해를 입힐 수 있음을 알 수 있다.

[3] IAM(Identity and Access Management): AWS 자원을 사용하도록 인증 및 권한 부여 등을 관리하는 서비스

[4] S3(Simple Storage Service): AWS 에서 제공하는 객체 스토리지 서비스

캐피탈 원에서 발생한 SSRF 공격 상세 시나리오는 다음과 같다.



① 공격자는 카드 디자인 변경 페이지의 파일 업로드 기능을 통해 URL 매개변수 확인
② 공격 대상 서버가 AWS 클라우드의 저장소인 S3 버킷을 사용하고 있는 것을 확인
③ SSRF 공격을 통해 공격 대상 서버의 AWS 메타데이터 서비스 접근
④ 공격자는 공격 대상 서버의 AWS Access Key, IAM 등 자격 증명 획득
⑤ 공격자는 획득한 정보를 바탕으로 AWS CLI 에 접근하여 내부 저장소(S3 버킷) 데이터 다운로드

※ 아래의 링크는 캐피탈원에서 발생한 SSRF 공격 내용을 재구성한 사이트이며, 관련 실습이 가능하다.
https://application.security/free-application-security-training/server-side-request-forgery-in-capital-one

이처럼 SSRF 공격에 성공할 경우 접근 권한이 없는 비인가자가 내부 서버의 자원에 접근할 수 있다. 따라서 메타데이터 API 를 사용하는 퍼블릭 클라우드를 대상으로 한 SSRF 공격이 증가하고 있는 추세이다. 2023 년 1 월, MS 의 클라우드 플랫폼인 Azure 의 4 가지 서비스에서 SSRF 취약점이 발견되어 긴급 패치가 이루어지기도 했다. 관련 취약점에 대한 자세한 내용은 아래의 링크를 통해 확인할 수 있다.

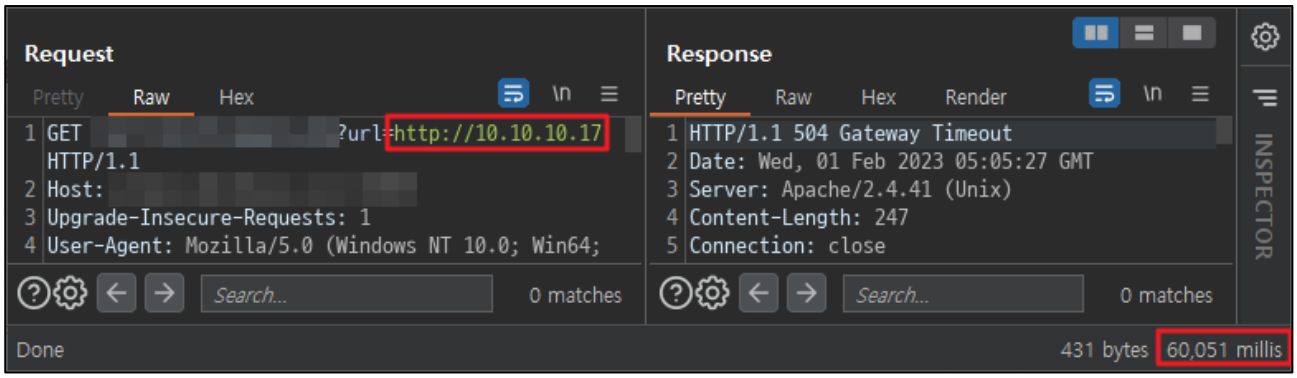
https://msrc-blog.microsoft.com/2023/01/17/microsoft-resolves-four-ssrf-vulnerabilities-in-azure-cloud-services/

3) 내부 네트워크 스캔

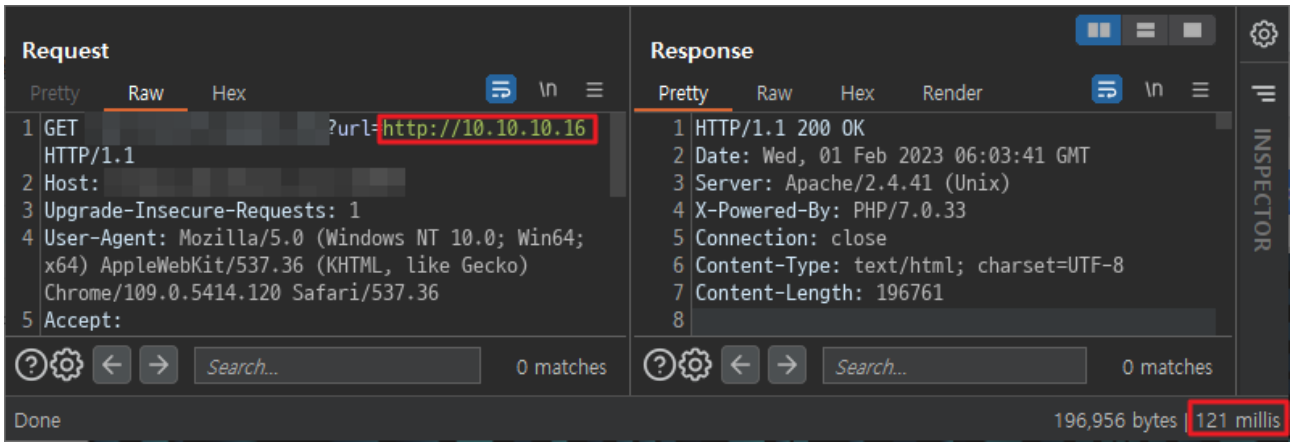
URL 뒤에 IP 대역 범위를 지정하여 요청하면 응답 시간, 길이, 데이터 등의 차이를 통해 내부 네트워크에 존재하는 포트 스캔이 가능하다. 이를 통해 Open 된 IP 와 Port 의 정보를 파악할 수 있다.

다음은 내부 서버에 존재하는 IP 를 스캔하는 예시이다. 대역 범위를 지정하여 하는 것이 일반적이지만, 실습은 특정 IP 를 알고 있다는 가정하에 진행되었다.

step 1) 내부 서버에 존재하지 않는 IP 인 10.10.10.17 에 대한 요청을 보냈을 때, 응답 시간이 60,051 millis(millisecond, 1/1000 초) 소요된 것을 확인할 수 있다.



step 2) 내부 서버에 존재하는 IP 인 10.10.10.16 에 대한 요청을 보냈을 때, 응답 시간이 121 millis 소요된 것을 확인할 수 있다. 이처럼 서버 측 응답 시간을 비교하여 내부 네트워크에 대한 정보 획득이 가능하다.

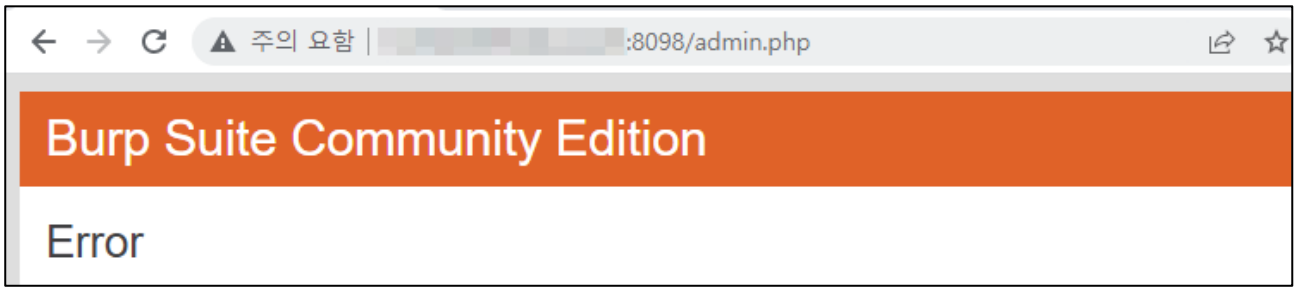


4) 외부 접근이 차단된 관리자 페이지 접근

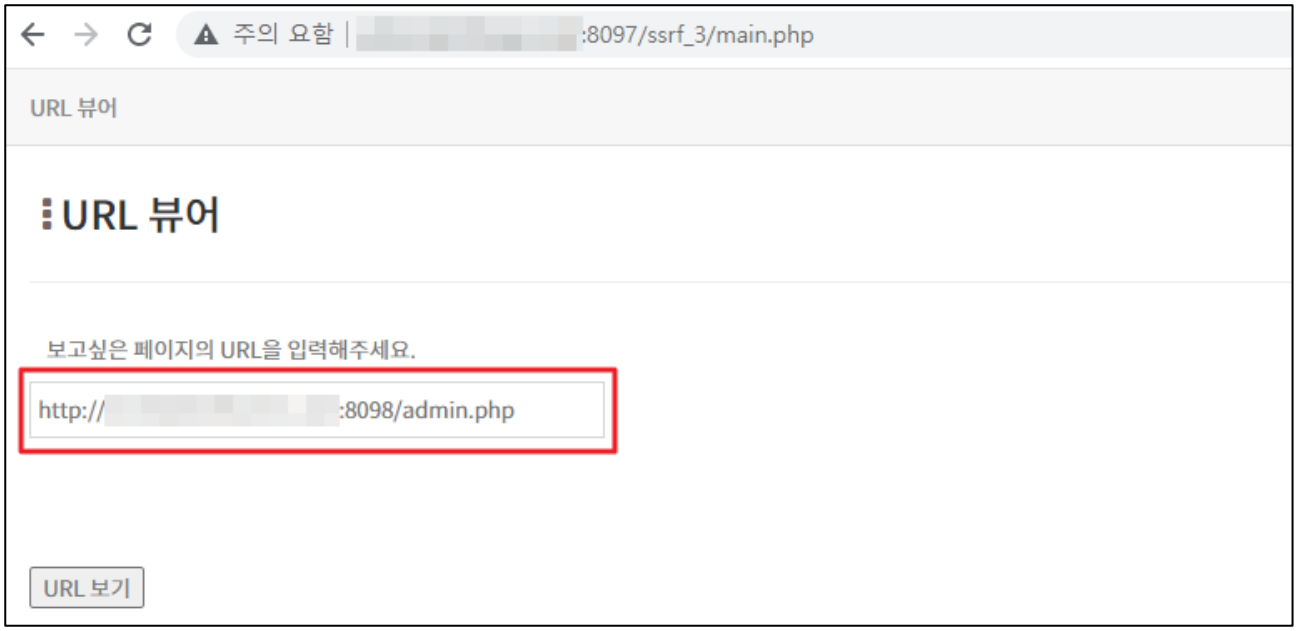
관리자 페이지는 허용된 IP 만이 접근할 수 있도록 설정하는 것이 일반적이지만, 이에 대한 설정이 미흡할 경우 SSRF 취약점을 통해 외부에서 관리자 페이지로의 접근이 가능하다.

다음은 내부 서버에 존재하는 관리자 페이지(admin.php)를 URL 뷰어를 통해 접근한 예시다.

step 1) 공격자는 내부 서버의 관리자 페이지 URL 을 획득하여 직접 접근하지만, 에러가 발생한 것을 확인할 수 있다.



step 2) 공격자는 해당 웹 서버의 접근 가능한 페이지에 존재하는 URL 뷰어를 활용하여 내부 서버의 관리자 페이지에 접근한다.



step 3) URL 뷰어를 통해 내부 서버의 관리자 페이지에 접근한 것을 확인할 수 있다.

■ 보안대책 및 우회기법

SSRF 는 사용자로부터 입력 받은 URL 파라미터에 대한 검증 미흡으로 서버 측 요청을 조작할 수 있는 공격이다. 기본적으로 사용자 입력값에 대해 검증하는 것이 중요하며, 특히 서버 측 요청에 사용되는 파라미터에 대한 검증에 신경 써야 한다. 따라서 서버 측에서 입력 받은 사용자의 데이터인 입력값을 WhiteList 기반으로 검증해야 한다.

1) WhiteList Filter

WhiteList Filter 방식은 사용자 입력값에 대한 요청을 허용할 List 를 작성하여, 해당 List 에 속하지 않을 경우 요청을 차단하는 방법으로 최우선으로 적용해야 하는 Filtering 방식이다.

서버에 접근하는 것을 허용할 URL 목록을 지정하여 List 에 등록되지 않은 URL 자체에 접근 권한이 없도록 설정해야 한다.

2) BlackList Filter

BlackList Filter 방식은 사용자 입력값에 대해 허용하지 않는 List 를 작성하여, 해당 List 에 속하는 요청일 때만 차단하는 방법이다. 서버에 접근을 허용하지 않는 URL 목록을 지정하여 List 에 등록된 URL 일 경우 접근을 차단하여 에러 페이지를 반환하는 등의 설정을 해야 한다. 하지만 BlackList Filter 방식을 사용하면 차단할 문자열 외에는 모두 허용하기 때문에 주의가 필요하다.

사용자 입력값에 대한 BlackList Filter 예시는 다음과 같다.

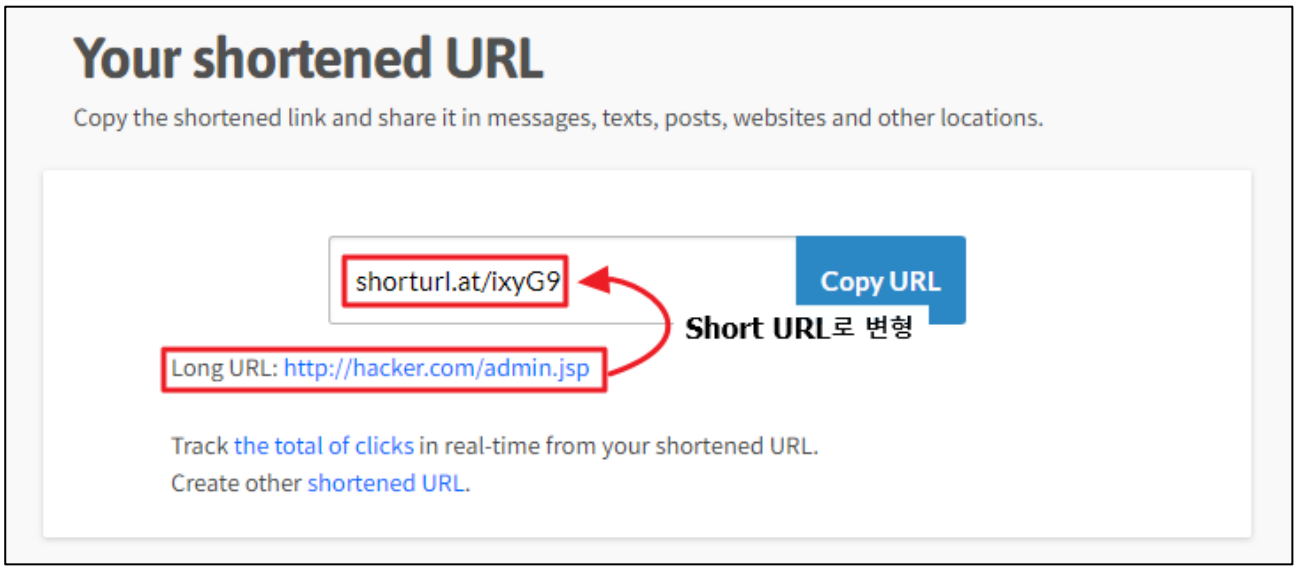
	입력값	입력값 예시
BlackList Filter 예시	사설 IP	10.0.0.0 - 10.255.255.255 172.16.0.0 - 172.31.255.255 192.168.0.0 - 192.168.255.255
	Loopback 주소	localhost, 127.0.0.1 등
	불필요한 Schema	sftp://, file://, http:// 등
	불필요한 특수문자	@, %0a 등

이처럼 BlackList 를 작성하여 접근을 차단할 사용자 입력값을 지정할 수 있지만, 다음과 같은 기법을 통해 우회가 가능하다.

- Short URL 기능을 이용한 우회 기법

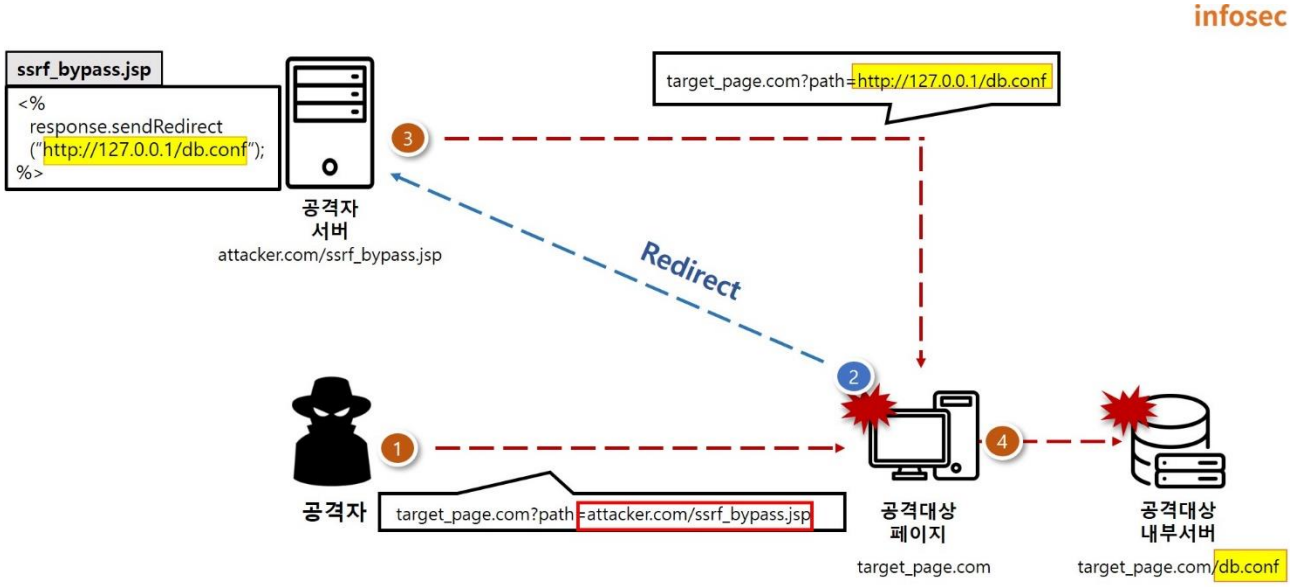
BlackList 로 지정한 URL 문자열에 대해 검증하기 때문에 URL 을 변형해주는 Short URL 기능을 이용하여 URL 문자열 검증을 우회할 수 있다.

서버 측에서 내부 서버의 관리자 페이지로 직접 접근이 불가능하도록 관리자 페이지 URL 을 BlackList 로 지정했을 경우, Short URL 을 통해 해당 문자열을 포함하지 않도록 변형할 수 있다.



- Redirect 기능을 이용한 우회 기법

공격자의 웹 서버에서 Redirect 시키는 페이지에 접근하도록 하여 BlackList 로 지정한 문자열 필터링을 우회하는 방법이다. 공격자의 서버에 공격 대상 서버의 페이지로 Redirect 하는 페이지를 만들어서 해당 URL 을 통해 공격 대상 서버에 접근하므로 BlackList Filter 를 우회할 수 있다.



이외에도 SSRF Filter 를 우회하는 다양한 방법이 존재하는데, 상세 내용은 아래의 링크에서 확인할 수 있다.

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Request%20Forgery

이처럼 우회 가능성이 있는 BlackList Filter 방식보다는 WhiteList Filter 를 통해 허용할 문자열을 지정하여 이에 대해서만 접근을 허가하는 방식 사용을 권장한다.

WhiteList Filter 를 통해 사용자 입력값에 대한 검증을 하는 것 외에 요청을 처리하는 서버와 중요 정보를 저장하는 내부 서버를 분리하여 운영하는 방법도 가능하다. 이 경우 요청을 처리하는 서버에서 SSRF 공격을 당하더라도 분리된 내부 서버에서 중요 정보가 유출되는 것을 방지할 수 있어 피해를 줄일 수 있다.

또한, 사용하지 않는 프로토콜이나 URL Schema 를 비활성화하고 네트워크 장비 등을 이용해 내부 서비스에 대한 접근 제한을 두는 것이 필요하다. 만약 특정 IP 를 기준으로 Filtering 한다면, 유효한 IP 인지에 대해 검증하는 라이브러리를 사용하여 외부로부터 받은 입력값에 의해 의도하지 않은 페이지로 이동이 불가능하도록 설정해야 한다.

■ 맺음말

2 월호에서는 서버 측 요청을 변조하여 공격자가 의도한 서버로 요청을 보내는 공격인 SSRF 에 대해 알아보았다. SSRF 취약점을 통 공격을 당할 경우, 외부에서 접근할 수 없는 내부 서버 또는 시스템으로 접근이 가능한 만큼 주의가 필요하다. 공격자는 조작된 HTTP 요청을 특정 서버로 전송할 수 있으며, 이는 내부 네트워크 정보를 파악하는데 이용될 수 있다. 특히, 내부 서버 및 시스템을 통해 탈취된 데이터가 중요 정보일 경우 공격 파급력이 더욱 높아질 수 있다.

SSRF 는 2022 년 OWASP(Open Web Application Security Project, 오픈 웹 애플리케이션 보안 프로젝트) 10 대 취약점에 포함되어 있으며, 클라우드로 환경이 전환되는 사례가 증가하고 있는 만큼 SSRF 공격은 앞으로도 계속될 것이다. 결정적으로 SSRF 는 본래 외부에서 접근할 수 없는 내부 서버에 접근할 수 있기 때문에 공격이 발생하지 않도록 주의해야 한다.

지금까지 EQST insight - Special Report 의 웹 취약점과 해킹 매커니즘 시리즈를 통해 SQL Injection, XSS(Cross-Site Scripting), CSRF(Cross-Site Request Forgery), SSRF(Server-Side Request Forgery)의 개념과 동작원리, 보안대책, 우회 기법 등의 내용을 다뤘다. 내부 시스템이나 웹페이지에 취약점이 존재하여 공격에 성공한다면, 클라이언트를 비롯해 서버에도 직접적인 영향을 미칠 수 있는 만큼 개발자는 취약점이 발생하지 않도록 개발해야 하고, 진단자는 취약점 진단 시 누락 없이 찾는 것이 중요하다.