Research & Technique

ProxyNotShell, Microsoft Exchange Server 원격 코드 실행 취약점 (CVE-2022-41040, CVE-2022-41082)

■ 취약점 개요

2022 년 8 월, 국내를 비롯해 전 세계 많은 기업에서 사용하는 메시징, 협업 소프트웨어 제품인 Microsoft 의 MS Exchange Server 에서 공격자가 유효한 메일 서버 계정을 이용해 시스템에 원격 코드 실행을 가능하게 하는 취약점이 발견됐다.

ProxyNotShell 이라 불리는 이 취약점은 MS Exchange Server 의 프론트엔드 서비스에서 제공하는 autodiscover[1] 서비스를 악용하여 유저가 직접 접근할 수 없는 백엔드의 파워 쉘을 이용하여 원격 코드 실행을 가능하게 하는 취약점이다.

인터넷상에서 공개된 MS Exchange Server 는 Shodan 과 같은 OSINT 검색 엔진을 통해 취약한 서버 운영 정보를 쉽게 확인할 수 있다. Shodan 검색 결과 2023-01-04 기준 전 세계에는 186,769 개의 MS Exchange Server 가 존재한다. 그리고 인터넷 보안 비영리 단체인 Shadowserver Foundation 은 ProxyNotshell 에 취약한 서버가 2023-01-02 기준 60,865 개 존재한다고 발표했다. 더욱이 최근 랜섬웨어 조직에서도 해당 취약점을 활용하고 있어 취약한 버전의 MS Exchange Server 사용 시 각별한 주의가 필요하다.

[1] MS Exchange Server autodiscover 프론트엔드란 Client access services 라고 불리며, Active Directory 도메인에 가입된 컴퓨터에게 내부/외부의 필요한 정보를 검색, 연결할 수 있는 서비스다. 외부에서 사용자들이 쉽게 서비스를 접근하도록 OWA(Outlook Web App)를 제공하는데 이를 autodiscover 프론트엔드라고 부른다.

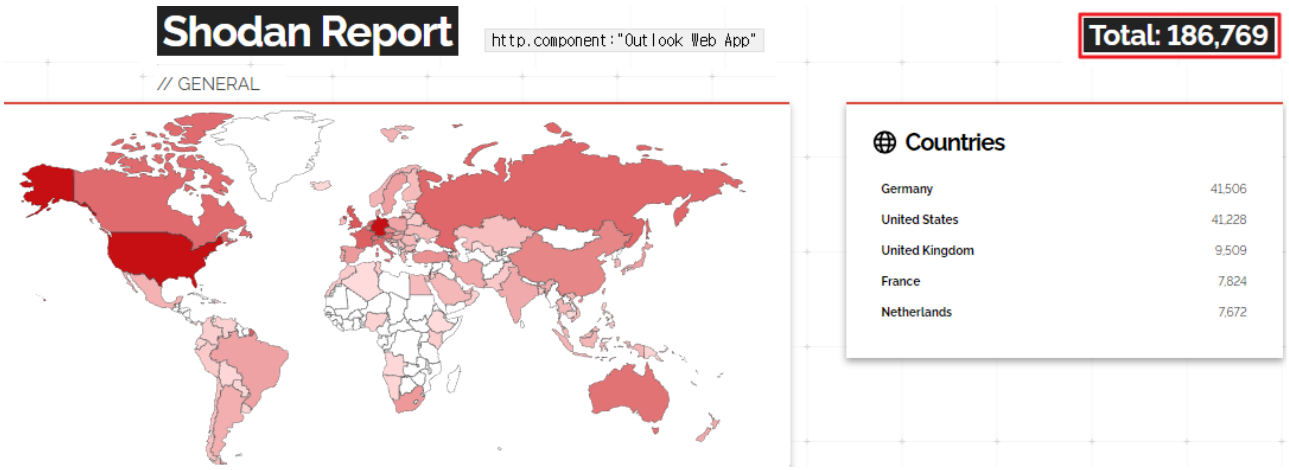


그림 1 전 세계 MS Exchange OWA(Outlook Web App) Server

■ 영향받는 소프트웨어 버전

ProxyNotShell 에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
MS Exchange Server	2013, 2016, 2019 version

※ 2022 년 11 월 8 일 이후 업데이트된 보안 패치를 적용하지 않은 서버는 취약하다.

■ 공격 시나리오

ProxyNotShell 를 이용한 공격 시나리오는 다음과 같다.

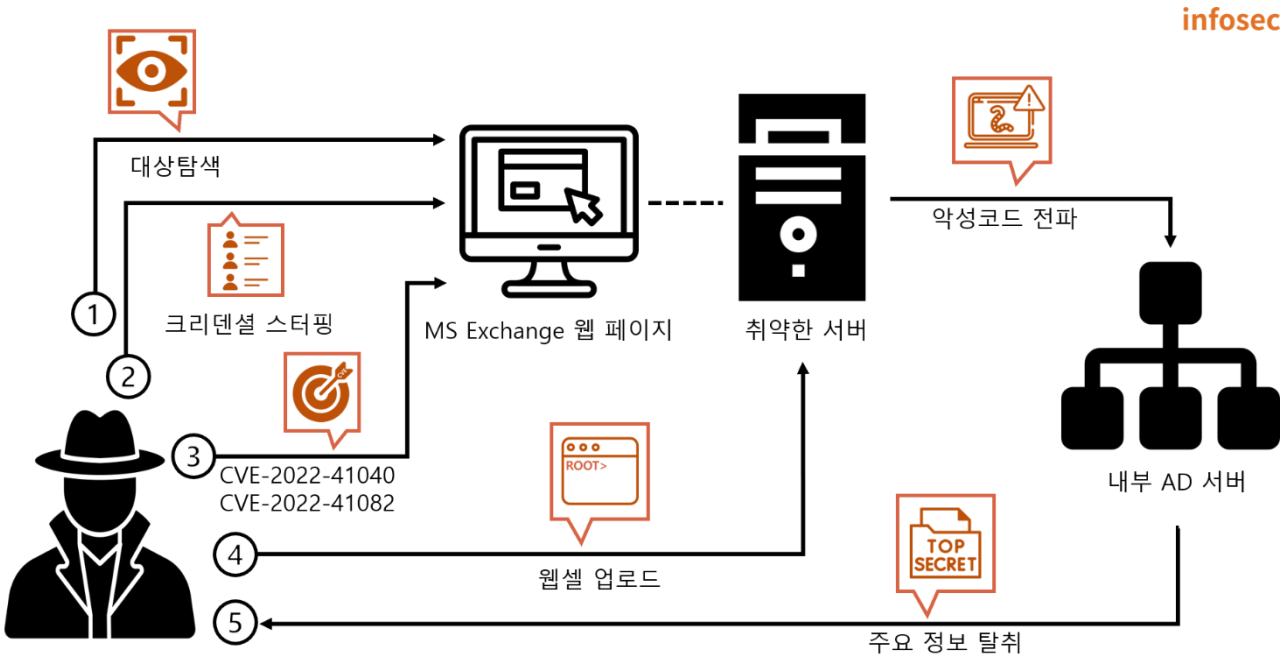


그림 2 공격 시나리오

① 공격자는 ProxyNotShell에 취약한 대상을 탐색
② 공격자는 크리덴셜 스터핑, 브루트 포싱, 피싱 등을 활용해 계정 정보 확보
③ 공격자는 MS Exchange Server 에 접속 가능한 사용자 계정을 통해 CVE-2022-41040(SSRF) 실행
④ 공격자는 CVE-2022-41082(RCE)를 악용한 백도어/웹쉘 설치, 악성코드 및 랜섬웨어 배포
⑤ 공격자는 피해자 PC 의 제어권 획득, 주요 정보 탈취 등 공격 수행

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 ProxyNotShell 의 동작 과정을 살펴본다.

이름	정보
피해자	Window Server 2012 R2 AD server (계정 정보: Administrator/EQST12#$) MS Exchange Server 2016 DNS (eqstlab.local)
공격자	Window 10 Pro AD user (계정 정보: user1/EQST12#$)

■ 취약점 테스트

Step 1. PoC 테스트

테스트를 위한 PoC 가 저장된 GitHub URL 은 다음과 같다.
- URL: https://github.com/testanull/ProxyNotShell-PoC

1) 다운로드 받은 PoC 코드를 활용하여 calc.exe 를 실행하도록 원격 코드 실행 시도

명령어	$ python poc_aug3.py https://eqstlab.local user1 EQST12#$ calc.exe $ python poc_aug3.py [도메인 경로] [ID] [Password] [명령어]

```
C:₩Users₩user1₩ProxyNotShell-PoC-main>python poc_aug3.py https://eqstlab.local user1 EQST12#$ calc.exe

[+] Create powershell session
[+] Got ShellId success
[+] Run keeping alive request
[+] Success keeping alive
[+] Run cmdlet new-offlineaddressbook
[+] Create powershell pipeline
[+] Run keeping alive request
[+] Success remove session
```

그림 3 RCE 명령 전송

2) 피해자(MS Exchange Server 2016) 서버에서 calc.exe 동작 확인

```
오후 3:28:23.4724772  cmd.exe        12760  CloseFile   C:₩Windows₩System32₩calc.exe          SUCCESS
오후 3:28:23.4727315  MSExchangeH... 13060  WriteFile   C:₩Program Files₩Microsoft₩Exchang... SUCCESS
오후 3:28:23.4792873  w3wp.exe       9576   Thread Exit                                        SUCCESS
오후 3:28:23.4803896  calc.exe       4912   Load Image  C:₩Windows₩System32₩calc.exe          SUCCESS
```

그림 4 RCE 동작 확인

■ 취약점 상세 분석

ProxyNotShell 은 CVE-2022-41040 과 CVE-2022-41082 의 연계 취약점이다.

Step 1. CVE-2022-41040

MS Exchange Server 는 사용자가 직접 접근 가능한 프론트엔드(Client access services)와 사용자가 직접 접근이 불가능한 백엔드(Backend Services)로 구성된다. ProxyNotShell 의 첫 번째 공격 단계는 CVE-2022-41040 을 활용한 SSRF [2] 취약점으로 OWA(Outlook Web App)의 autodiscover 프론트엔드를 악용한다. autodiscover 서비스란 API 엔드포인트를 이용하여 내부 서버에 필요 기능을 요청하는 서비스다. 해당 취약점은 autodiscover 프론트엔드에서 PowerShell API 엔드포인트 URL 경로를 PowerShell 접근 경로로 변경하여 접근 불가능한 백엔드의 액세스 권한을 획득한다.

※ PowerShell API 엔드포인트 경로는 https://%exchangeserverdomiaon%/powershell 이다.

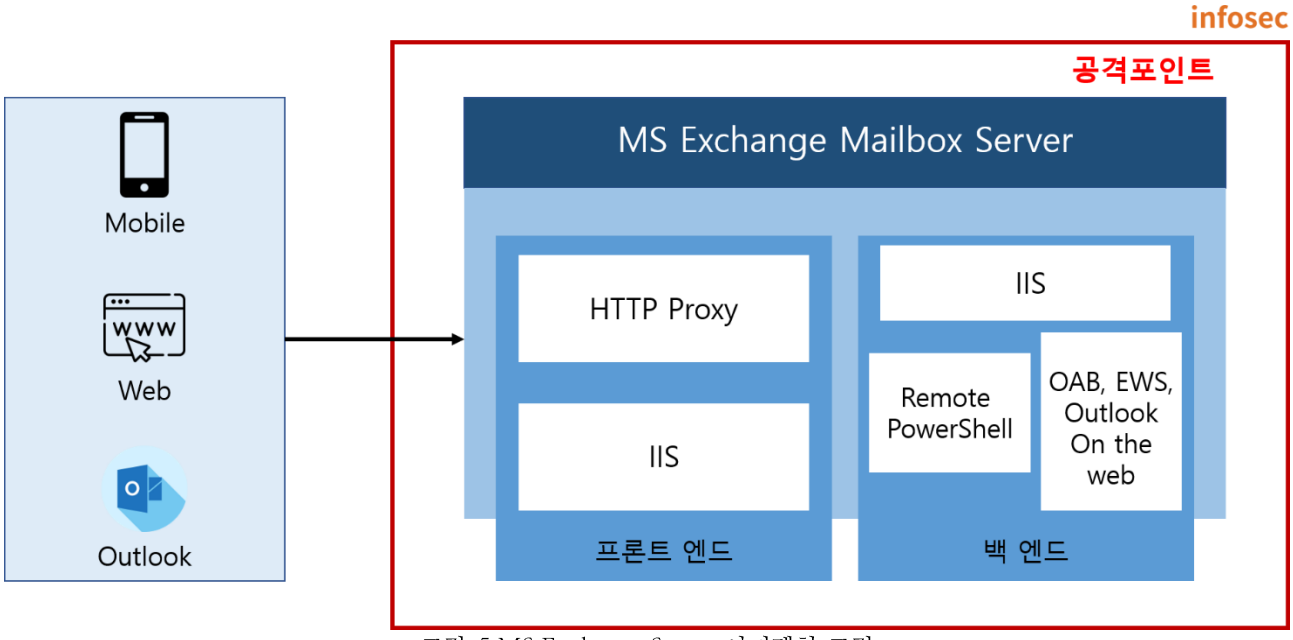


그림 5 MS Exchange Server 아키텍처 그림

[2] SSRF(Server-Side Request Forgery)의 약자로 서버 측에서 위조된 요청을 보내도록 하는 취약점으로, 내부 서버의 정보를 노출시키거나 서비스 거부 공격을 할 수 있다.

CVE-2022-41040 취약점은 2021 년 공개된 MS Exchange Server 취약점인 ProxyShell(CVE-2021-34473, SSRF)의 보안대책을 우회한 취약점이다.

- ProxyShell 란?

CVE-2021-31207(MS Exchange Server 보안 기능 우회 취약점), CVE-2021-34523(MS Exchange Server 권한 상승 취약점), CVE-2021-34473(MS Exchange Server RCE 취약점)의 연계 취약점으로 인증되지 않은 사용자가 원격 코드 실행이 가능한 취약점이다.

ProxyShell 취약점은 autodiscover 를 이용하여 URL 을 검색할 때, 필터링이 미흡한 백엔드 URL 을 악용하여 백엔드로 접근하는 SSRF 취약점이다. 2021 년 7 월에 인증 요소를 추가함으로써 백엔드의 액세스를 제한하여 악용할 수 없도록 패치 되었다. ProxyShell 의 보안패치가 적용된 서버에서 공격을 시도해 본 결과, 그림 6 과 같이 응답 값으로 401 Unauthorized 에러를 반환한다. 하지만, 그림 6 처럼 응답 값 부분에 인증 방식 NTLM(NT Lan Manager)[3]과 Basic[4]이 노출되어 이를 이용한 공격이 가능하다.

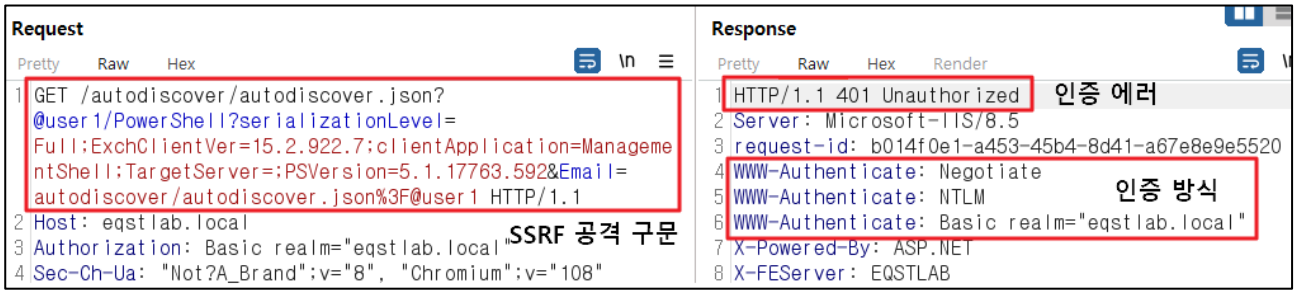


그림 6 ProxyShell 401 인증오류

[3] NTLM 이란 NT Lan Mange 의 약어로, 윈도우에서 제공하고 있는 인증 프로토콜 중 하나로 Challenge-Response 의 인증 프로토콜 방식으로 암호를 전송하지 않고, Exchange 호스트에 연결할 수 있다.

[4] Basic 이란 웹 서버에 보낼 아이디와 암호를 Base64 방식으로 인코딩 후 전달하여 인증하는 방식으로 Basic 인증을 이용해 EWS(Exchange Web Service), 원격 PowerShell, autodiscover 등을 사용할 수 있다.

따라서 SSRF 페이로드를 포함하여 ProxyNotShell 공격을 위해 그림 7 처럼 유효한 인증 정보인 user1:EQST12#$를 base64 로 인코딩하여 Request 를 보낸다.

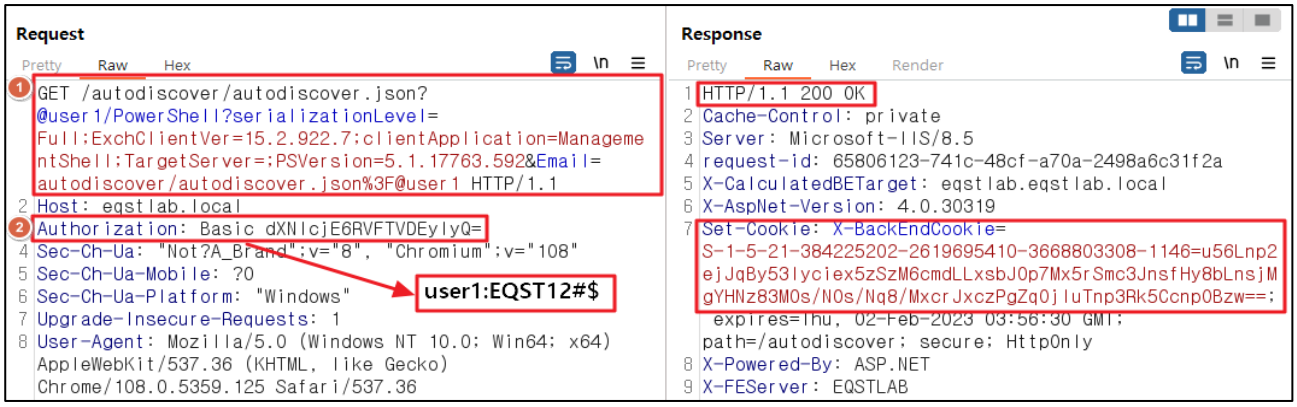


그림 7 ProxyNotShell 이용 PowerShell 내부 접근

유효한 인증 정보를 헤더에 추가하면 ProxyShell 보안대책이 우회 가능하여, autodiscover 를 악용한 SSRF 가 여전히 동작하여, 접근 권한이 저장된 백엔드의 쿠키 값을 얻을 수 있다.

Step 2. CVE-2022-41082

Exchange PowerShell 백엔드 접근에 성공했으므로, PowerShell Remoting5 에서 역직렬화 취약점을 악용해 원격 코드 실행이 가능한 취약점인 CVE-2022-41082 를 실행한다. 이 공격에는 직렬화(Serialization)와 역직렬화(Deserialization)가 사용된다. 직렬화란 객체를 네트워크를 통해 다른 곳으로 전송할 수 있는 형식이나 파일에 저장할 수 있는 데이터인 바이트 또는 스트림으로 변환시키는 것을 의미한다. 역직렬화란 직렬화한 데이터를 본래의 객체로 되돌리는 변환을 의미한다. CVE-2022-41082 는 공격자가 악의적으로 조작된 직렬화 데이터를 저장한 뒤, 역직렬화 과정에서 원격 코드 실행이 가능한 취약점이다.

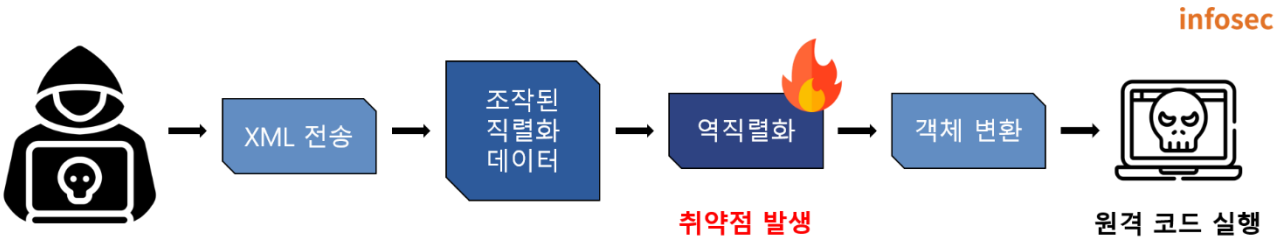


그림 8 CVE-2022-41082 역직렬화 취약점 동작 과정

[5] PowerShell Remoting 이란 사용자가 원격으로 PowerShell 명령을 실행할 수 있는 서비스이다.

Step 3. PoC 동작 분석

아래의 PoC 는 SSRF 를 통해 공격을 성공한 뒤, 객체를 루트에 반환하는 메소드인 XamlReader 를 통한 역직렬화 과정에서 페이로드를 객체로 Xaml 을 로드하여 RCE 를 실행하는 코드의 일부이다.

그림 9 PoC RCE 주요 페이로드

역직렬화 허용을 위해 1 번과 같이 TargetTypeForDeserialization 을 설정하고, 2 번과 같이 System.Exception[6] 로 Type 유형을 선언한다. Type 을 Exception 으로 설정한 이유는 취약한 BinaryFormatter 를 포함된 클래스인 SerializationTypeConverter 를 활용하면 RCE 페이로드 부분을 악용할 수 있기 때문이다. MS Exchange Server 에서 객체를 직렬화 할 때는 UnitySerializationHolder[7] 클래스를 활용한다. 따라서, 〈MS〉태그(4 번)에 객체를 직렬화 할 때 필요한 UnitySerializationHolder 와 XamlReader 를 인코딩하여 포함시켜 직렬화하면, 역직렬화 될 때 RCE 페이로드 문자열이 저장된 〈S〉태그를 객체로 활용할 수 있다.

[6] System.Exception 클래스는 애플레이케이션 실행 중 발생하는 예외를 다루기 위한 클래스로 .NET 에 포함되어 있다. https://learn.microsoft.com/ko-kr/dotnet/api/system.exception?view=net-7.0 에 정의되어 있다.

[7] UnitySerializationHolder 클래스는 라이브러리 "mscorlib"에 포함되어 있으며, MS Exchange Server 에서 직렬화를 다룰 때 사용하는 클래스이다.

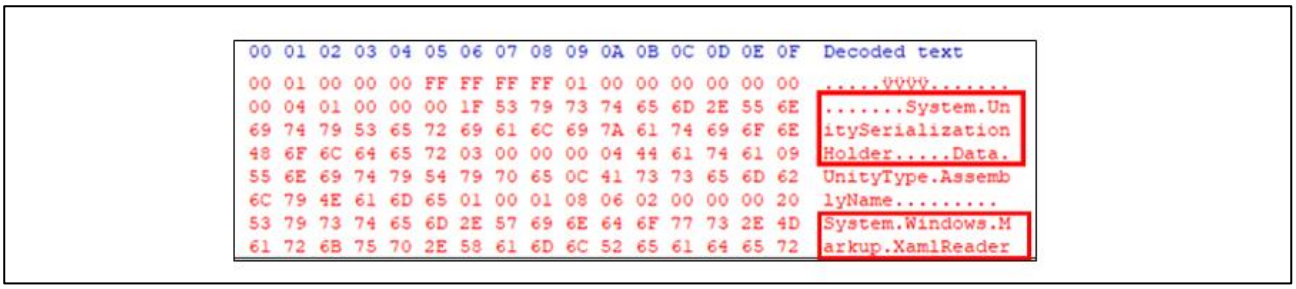


그림 10 디코딩 된 값

정리하자면 UnitySerializationHolder 와 XamlReader 를 직렬화하면, 역직렬화 시 XamlReader 에서 페이로드를 객체로 사용하여 실행하고, Xaml 이 로드되어 원격 코드 실행이 가능하다.

```
  <TypeConverter>
    <TypeName>Microsoft.Exchange.Data.SerializationTypeConverter</TypeName>
  </TypeConverter>
</Type>
<Type>
  <Name>Deserialized.Microsoft.Exchange.Data.Unlimited`1[[Microsoft.Exchange.Data.EnhancedTimeSpan, Microsoft.Exchange.Data,
       Version=15.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35]]</Name>
  <Members>
    <MemberSet>
      <Name>PSStandardMembers</Name>
      <Members>
        <NoteProperty>
          <Name>TargetTypeForDeserialization</Name>
          <Value>Microsoft.Exchange.Data.Unlimited`1[[Microsoft.Exchange.Data.EnhancedTimeSpan, Microsoft.Exchange.Data,
               Version=15.0.0.0, Culture=neutral,  PublicKeyToken=31bf3856ad364e35]]</Value>
        </NoteProperty>
      </Members>
```

그림 11 xml 의 등록된 모습

역직렬화 시 페이로드의 {CMD}에 calc.exe 가 객체로 전달되어 cmd.exe /c calc.exe 가 완성되어 RCE 가 실행된다.

```
<![CDATA[<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:System="clr-namespace:System;assembly=mscorlib"
 xmlns:Diag="clr-namespace:System.Diagnostics;assembly=system"><ObjectDataProvider x:Key="LaunchCalch"
 ObjectType="{{x:Type Diag:Process}}" MethodName="Start">
 <ObjectDataProvider.MethodParameters><System:String>cmd.exe</System:String><System:String>/c {CMD}
 </System:String> </ObjectDataProvider.MethodParameters> </ObjectDataProvidr> </ResourceDictionary>]]>
```

그림 12 전달객체

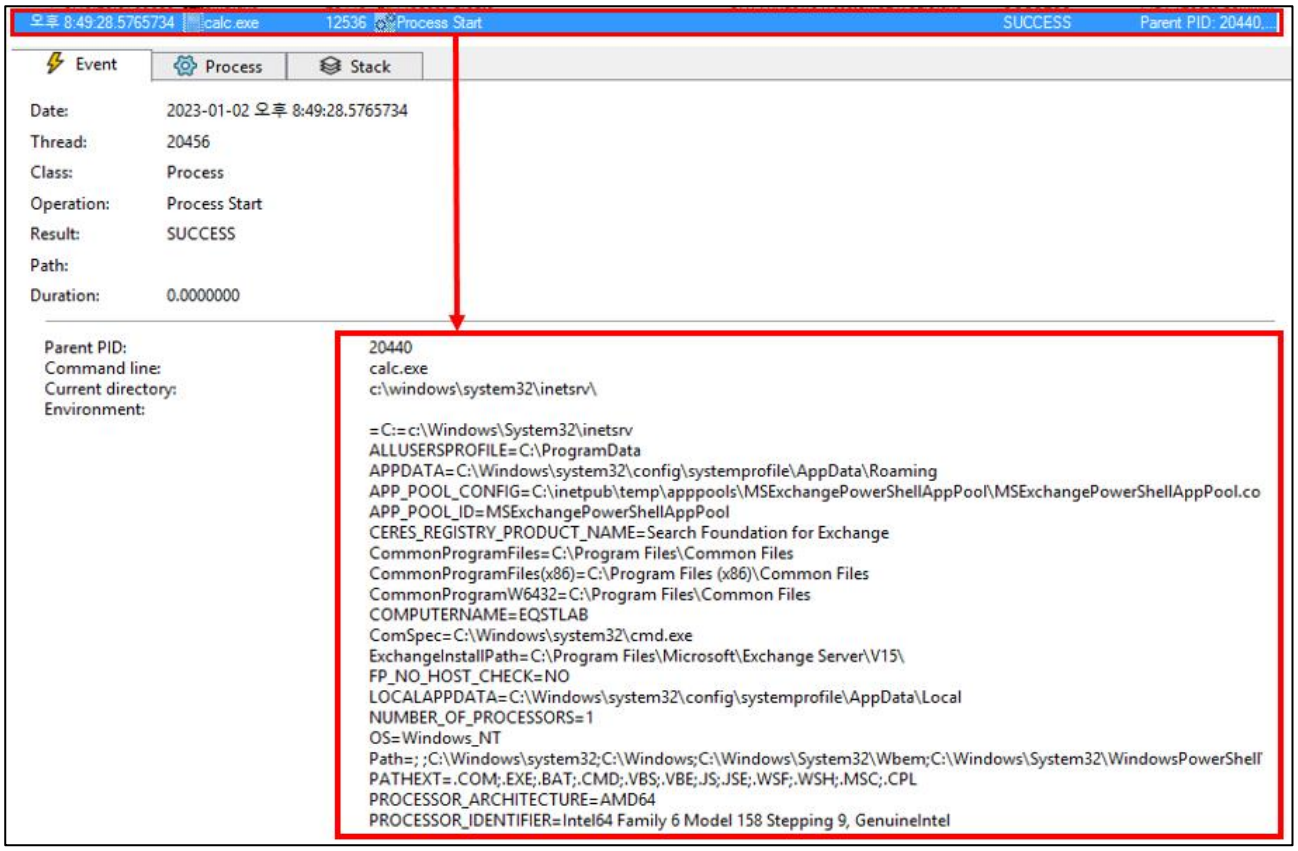


그림 13 RCE 실행 결과 모습

다음은 Python PoC 의 동작 과정의 설명이다. RCE 를 실행하기 위해서 MS Exchange Server 주소, ID, 비밀번호와 실행할 명령어를 인자 값으로 전달하면, 주소를 기반으로 autodiscover 를 통해 PowerShell 엔드 포인트 URL 에 base64 로 인코딩 된 ID 와 비밀번호를 전달하여 SSRF 를 진행한 뒤, 그림 13 의 {CMD}에 실행할 명령어가 객체로 전달되어 원격 코드 실행이 가능하다.

■ 공격 변화 양상

최근 MS Exchange Server 관련 취약점들이 다양하게 발생하고 있다. 아래의 표는 2021 년 발생한 ProxyLogon부터 최근 공격자들 사이에서 많이 사용되는 OWASSRF에 대한 내용을 정리한 표이다.

취약점	특징
ProxyLogon	• CVE-2021-26855(SSRF 취약점) + CVE-2021-27065(Arbitrary File Write 취약점) • 2021 년 1 월 발견 • 인증되지 않은 사용자가 외부 서버를 이용하여 내부 서버로 실행가능 파일 저장 가능 • 대응방안 - 인증 로직 추가 및 파일 확장자 검사 로직 추가, KB5000871 패치 적용
ProxyShell	• CVE-2021-34473(MS Exchange Server RCE 취약점) + CVE-2021-34523(MS Exchange Server 권한 상승 취약점) + CVE-2021-31207(MS Exchange Server 보안 기능 우회 취약점) • 2021 년 4 월 발견 • autodiscover 를 활용한 백엔드에 접근 후 임의 파일 쓰기를 통한 RCE 실행 • 대응방안 - 파일 확장자 제한, 인증 로직 추가, KB5004778 패치 적용
ProxyNotShell	• CVE-2022-41040 (Auto Discover SSRF 취약점) +CVE-2022-41082 (RCE 취약점) • 2022 년 8 월 발견 • autodiscover 를 활용한 백엔드 접근 후 RCE 실행 • 대응방안 - URL 차단 규칙 추가, 비 관리자를 위한 원격 PowerShell 비활성화, KB5019758 패치 적용
OWASSRF	• CVE-2022-41080(OWA SSRF 취약점) + CVE-2022-41082(RCE 취약점) • 2022 년 11 월 발견 • OWA 엔드포인트를 통한 백엔드 접근 후 RCE 실행 • 대응방안 - OWA 비활성화, 원격 PowerShell 제한, KB5019758 패치 적용,

취약한 MS Exchange Server 를 노린 공격은 2021 년 발견된 ProxyLogon 이후 더욱 활발해졌다. ProxyLogon 은 인증되지 않은 사용자가 외부 서버를 통해 내부 서버에 Serverside 언어(.aspx)를 저장한 후 실행 가능한 취약점이다.

이후 MS Exchange Server 관련 취약점 연구가 증가했으며, 같은 해 보안 기능 우회/권한 상승/원격 코드 실행 취약점을 연계한 ProxyShell 이 등장했다. 2021 년 5 월~7 월에 각 취약점에 대한 패치가 진행되었지만, 2022 년에 해당 패치를 우회한 ProxyNotShell 이 등장했다.

ProxyNotShell 역시 MS 에서 발표한 긴급 보안 대책을 우회한 OWASSRF 로 발전해 현재 Play 랜섬웨어 집단에서 공격 방법으로 사용하고 있다. OWASSRF 는 OWA(Outlook Web Access) 엔드포인트를 통해 이뤄지며 /owa/<email_address>/ProwerShell 과 같이 OWA URL 사서함을 활용한 뒤, ProxyNotShell 의 CVE-2022-41082(RCE)를 실행한다.

2022 년 발견되어 현재까지 실제 공격에 이용되고 있는 ProxyNotShell 과 OWASSRF 의 차이점은 SSRF 공격에 사용되는 프론트엔드의 엔드포인트 공략 지점에 있다. ProxyNotShell 은 autodiscover 를 사용하고, OWASSRF 는 OWA 엔드포인트를 활용한다. 엔드포인트를 통한 백엔드 접근 이후 원격 코드 실행 취약점인 CVE-2022-4182 를 활용하는 점은 동일하다. 다음은 ProxyNotShell 과 OWASSRF 의 차이점을 나타낸 그림이다.

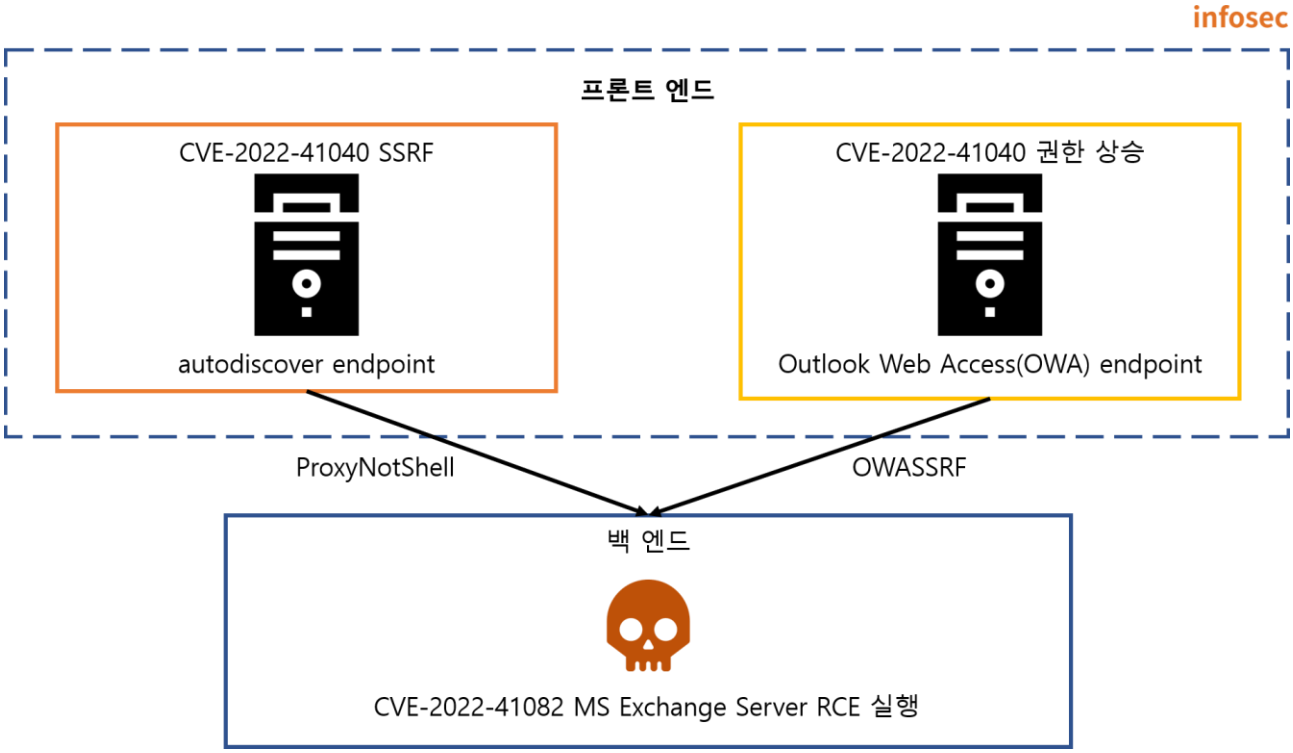


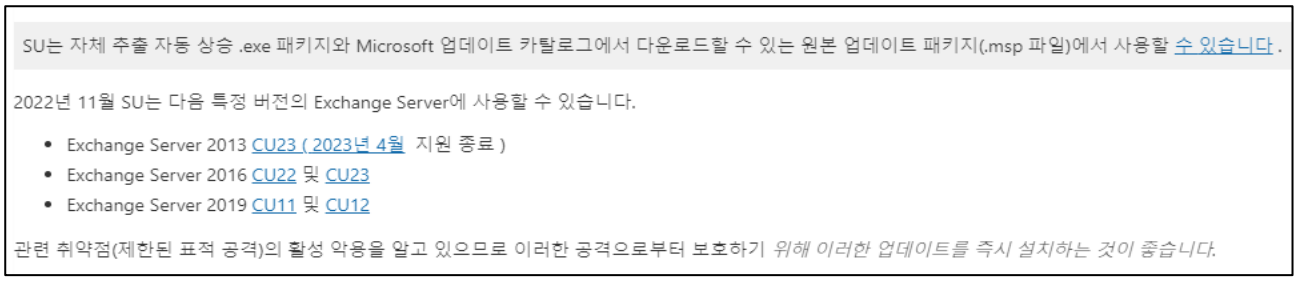
그림 14 ProxyNotSHell VS OWASSRF

■ 대응 방안

취약점 최초 발견 당시 Microsoft 에서는 URL 차단 규칙을 수동으로 설정하는 것을 권고했다. 하지만 이 방법은 우회 가능성으로 인해 효과적이지 않으며 새로 발생한 취약점인 OWASSRF 를 방지할 수 없는 문제가 있다.

따라서 취약점에 대비하기 위해 2022-11-8 이후 업데이트된 패치(KB5019758)를 적용해야 한다. 업데이트된 패치에 대한 내용은 아래의 링크에서 확인할 수 있다.

https://techcommunity.microsoft.com/t5/exchange-team-blog/released-november-2022-exchange-server-security-updates/ba-p/3669045


SU는 자체 추출 자동 상승 .exe 패키지와 Microsoft 업데이트 카탈로그에서 다운로드할 수 있는 원본 업데이트 패키지(.msp 파일)에서 사용할 수 있습니다 .

2022년 11월 SU는 다음 특정 버전의 Exchange Server에 사용할 수 있습니다.

- Exchange Server 2013 CU23 (2023년 4월 지원 종료)
- Exchange Server 2016 CU22 및 CU23
- Exchange Server 2019 CU11 및 CU12

관련 취약점(제한된 표적 공격)의 활성 악용을 알고 있으므로 이러한 공격으로부터 보호하기 *위해 이러한 업데이트를 즉시 설치하는 것이 좋습니다.*

그림 15 MS Exchange Server 패치 내역

만약 불가피하게 업데이트를 할 수 없는 경우 패치 적용 전까지 관리자가 아닌 사용자가 PowerShell 를 원격으로 실행하지 못하게 원격 PowerShell 를 비활성화해야 한다. 또한, ProxyNotShell 의 우회 취약점인 OWASSRF 에 대비하기 위하여 OWA 를 비활성화해야 한다. 웹 방화벽이나 EDR(Endpoint Detection & Response) 도구를 활용하여 악의적인 요청(PowerShell 실행, cmd 실행 등)을 탐지해야 하며, X-Forwarded-For 헤더가 프록시 서비스에 대한 요청을 위해 실제 외부 IP 주소를 기록하도록 구성되었는지 확인해야 한다. 그럼에도 임시 대책으로 우회할 가능성이 있으므로, 업데이트된 패치를 상시 확인 후 적용해야 안전하다.

■ 참고 사이트

• URL: https://www.crowdstrike.com/blog/owassrf-exploit-analysis-and-recommendations/
• URL: https://www.zerodayinitiative.com/blog/2022/11/14/control-your-types-or-get-pwned-remote-code-execution-in-exchange-powershell-backend
• URL: https://www.rezilion.com/blog/proxyshell-or-proxynotshell-lets-set-the-record-straight/
• URL: https://twitter.com/wdormann/status/1593630153036500993/photo/1
• URL: https://www.shodan.io/search/report?query=http.component%3A%22Outlook+Web+App+%22