Detecting a ZeroTier Backdoor

Kyle Desjardins
09 July 2024

I recently created a proof-of-concept where I explored the possibility of using a combination of ZeroTier, a PowerShell script and a scheduled task as part of some post-exploitation activities to install persistence on a windows computer. The link to that video and the repo is on the last page of this paper.

The logical flow of the attack is as follows:

1. Place the .msi installer for ZeroTier somewhere on the system
2. Create a scheduled task to periodically run a PowerShell script that:
    - Installs ZeroTier by running the .msi installer
    - Creates a new local user, sets their password, and adds them to the administrator group
    - Joins the attacker controlled ZeroTier Network
    - Waits to see if the attacker connects
        - If connection occurs, stay on the ZeroTier Network until the attack decides they're finished on the victim machine
        - If no connection is created within 20 seconds, or the attacker is finished; leave the ZeroTier Network
    - Deletes the user that was created
    - Uninstalls ZeroTier

Since this test was successful, I decided that it would be important to try gathering all of the potential indicators that defenders could use in the event that a malicious actor was using a similar technique on their network. Before getting into the methodology of gathering these indicators and describing the indicators themselves, what is ZeroTier anyway?

What is it and why it's scary

ZeroTier is a software-defined networking service that enables the creation of secure, virtual networks over the internet. It combines the features of VPNs and SD-WANs, allowing devices to communicate as if they were on the same local network, regardless of their physical location. Key features include:

- **Peer-to-Peer Networking:** Establishes direct connections between devices to optimize speed and reduce latency.
- **Cross-Platform Support:** Available for various network appliances such as switches, as well as operating systems, including Windows, macOS, Linux, Android, and iOS
- **Ease of Use:** Simplifies network configuration, requiring minimal setup and management.
- **Encryption:** Uses strong encryption (AES-256) to secure data transmissions.

How ZeroTier Can Be Abused by Malicious Actors:

- **Unauthorized Network Access**
  - If a malicious actor gains access to a ZeroTier network, they can potentially access all devices connected to that network. This access can be exploited to steal sensitive data, deploy malware, or conduct further attacks within the network.
- **Bypassing Firewalls and Network Segmentation**
  - ZeroTier can create connections that bypass traditional network security measures such as firewalls and network segmentation. An attacker could use this to move laterally within an organization's network, reaching systems that would otherwise be protected.
- **Disguising Malicious Traffic**
  - Because ZeroTier encrypts traffic and routes it through its network, it can be used to obscure malicious activity from network monitoring tools. This can help attackers avoid detection while exfiltrating data or communicating with compromised systems.
- **Persistence Mechanism**
  - Attackers can use ZeroTier as a persistent backdoor to maintain access to a compromised network. Even if other malicious software is detected and removed, the ZeroTier connection can allow attackers to regain entry.
- **Evasion of Geo-Restrictions and Content Filtering**
  - Malicious actors can use ZeroTier to evade geographic restrictions and content filters, facilitating activities like accessing illegal content or conducting attacks from jurisdictions with less stringent cybercrime laws.

Methodology for Gathering Indicators

First, I cleared all the event logs on my test system to reduce the amount of noise from other applications or services running on the system during or before my test. I took a registry snapshot prior to doing anything and began a Wireshark capture on my default LAN interface. After the installation of ZeroTier, the newly created ZeroTier virtual tunnel interface was available, and I began Wireshark capture on that interface as well.

Summary of Data sources

- Evtx and Filesystem
- Registry snapshot
- Pcap

I proceeded to run through the attack outlined at the beginning of this paper. I decided not to bother with the scheduled task, that's just what I chose to use because it was easy for my proof-of-concept. An attacker has no obligation to use a scheduled task. For that reason, I chose to

focus on the ZeroTier application with no regard to the method of running it. I wanted to take an agnostic approach that would help defenders determine if a machine on their network is using ZeroTier as a backdoor/persistence mechanism.

I installed ZeroTier and joined a ZT network through API Calls while running a pcap and took note of all the filesystem hashes, registry modifications, and important event logs that resulted as part of this process.

ZeroTier has its default configuration, but also the ability to deviate from the defaults. That's an important thing to bring up because when it comes to indicators, some may be more reliable than others. The Default configuration provides more atomic indicators than the plethora of non-default configuration possibilities. For the non-default configurations, the detections are behavioral based. To further complicate things, there can absolutely be a mix of default and non-default configurations, for example, UDP port 9993 is not used but the name of driver is still zttap, etc.

Indicators - The default configuration

> Note: the term "default interface" is used to indicate the network interface on the victim device that has the normal DHCP or statically assigned IP address, specifically NOT the virtual ZeroTier tunnel interface that is created by the application. I've chosen to not include specific indicators noticed in the traffic on the ZeroTier Tunnel Interface because you would never see that traffic in a SIEM and capturing that traffic on a per-host basis is not really feasible.

- UDP destination port 9993 traffic to ZeroTier registered IP addresses from the default interface
  - Name: root-mia-01.zerotier.com - Address: 103.195.103.66
  - Name: root-zrh-01.zerotier.com - Address: 84.17.53.155
  - Name: root-sgp-01.zerotier.com - Address: 50.7.252.138

- DNS queries for ZeroTier URLs from the default interface
  - my.zerotier.com
  - *.zerotier.*

- DNS queries for the following from the default interface
  - _nomachine._tcp.local
  - dejavu._nomachine._tcp.local

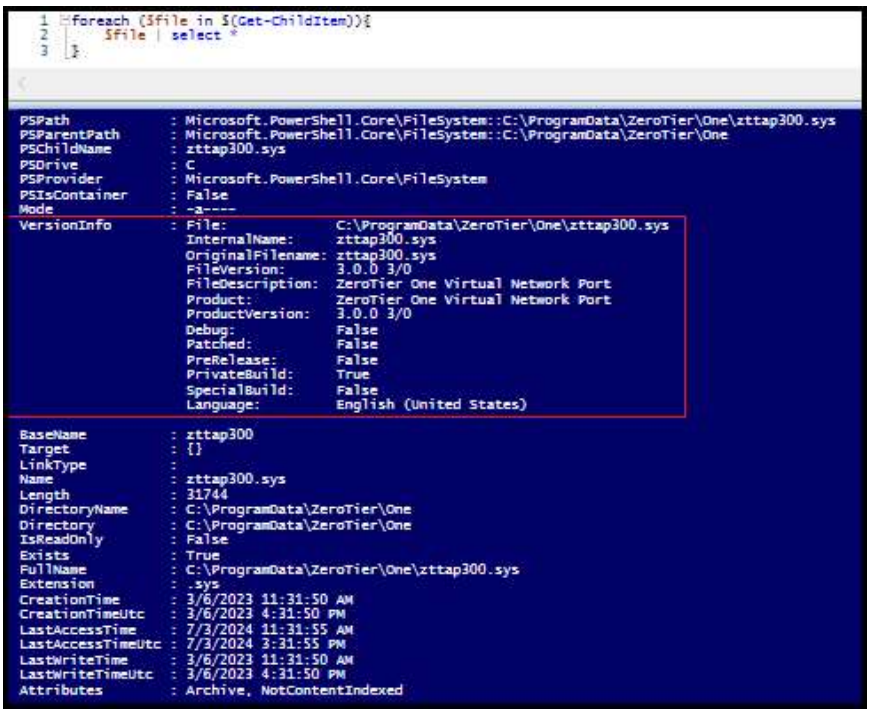
- ZeroTier pre-defined IPv4 address pools to assigned to devices on the network
    - Obtain a listing of the network interfaces from each device on your network by running an ipconfig, or other equivalent method and check to see if there are any interfaces present that have any of the following IP address prefixes assigned to them.

ZeroTier Default IP Prefixes			
10.147.17.*	10.147.18.*	10.147.19.*	10.147.20.*
10.144.*.*	10.241.*.*	10.242.*.*	10.243.*.*
10.244.*.*	172.22.*.*	172.23.*.*	172.24.*.*
172.25.*.*	172.26.*.*	172.27.*.*	172.28.*.*
172.29.*.*	172.30.*.*	192.168.191.*	192.168.192.*
192.168.193.*	192.168.194.*	192.168.195.*	192.168.196.*

- The term “ZeroTier” present in filenames, file metadata, and registry Keys on the victim computer

Filesystem

Assuming that the file names/paths and hashes can potentially change it was also noticed that there’s some ZeroTier metadata that might be left behind in the properties of each file. This of course would require some type of enumeration script to be run per-host, but the info is there, nonetheless.



Filesystem		
Type	**File**	**SHA256**
Filesystem	C:\ProgramData\ZeroTier\One\zttap300.cat	38C0BBBCA2E32509A32150990DBEBF98BE4FFE2C141A23E878CBEB6C7D483365
Filesystem	C:\ProgramData\ZeroTier\One\zttap300.inf	4AFB5E5A2897350926704CDCB427FD69BD2C9F0DA271941D1484896B54D294AE
Filesystem	C:\ProgramData\ZeroTier\One\zttap300.sys	4221486A0D36EF96AB1ED2A537747388655C4C2E470911646F4102D0B88FBDC6
Filesystem	C:\ProgramData\ZeroTier\One\metrics.prom	F9E2F121A3DDADD784442FF2B7D9BE8FA1288E1F97D0A1762CDE770250143F2E
Filesystem	C:\Program Files (x86)\ZeroTier\One\regid.2010-01.com.zerotier_ZeroTierOne.swidtag	60E33E3DDC578AB8CA5B27DFA59DE4326F5700E03FB9C01A2DEF23B051BF3E3C
Filesystem	C:\Program Files (x86)\ZeroTier\One\zerotier-cli.bat	211E5E8421AA5CBF5FCFD8548AE7A5AFF6F3360765F122B547A693D88A261BC6
Filesystem	C:\Program Files (x86)\ZeroTier\One\zerotier-idtool.bat	21DFCFD3191E8F74AF099C0EC50C01167377FE844A011B47E2D50DAFE62779A1
Filesystem	C:\Program Files (x86)\ZeroTier\One\zerotier_desktop_ui.exe	9F928566B9A728C0E994DDA59254E26E17060FEE03AE0AED1D5AB6F96A64D74C
Filesystem	C:\ProgramData\ZeroTier\One\zerotier-one_x64.exe	69006814689EC24EFC2BC8E5E53542F3D9FDA277D86EDC083E81473C08B510E8

Registry	
Type	Path
Registry	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Setup\PnpLockdownFiles\%SystemRoot%/System32/drivers/zttap300.sys
Registry	HKLM\SYSTEM\ControlSet001\Services\zttap300

The string “ZeroTier” anywhere in any of the properties or subkeys of either of the following registry locations:

- HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall

Event Logs

The event logs left a lot to be desired. Most of the logs that were generated and confirmed to be a direct result of either the ZeroTier Install, the virtual tunnel interface, or any other aspect of connecting to the victim machine, lack the specificity to pinpoint the attack. Furthermore, very few of them had atomic indicators. The events that have the term “ZeroTier One” somewhere in the message field, likely obtained this product data from the .msi installer file, so if the installer has been sanitized of this metadata, I can’t say for sure that this term would be present in the logs.

- Application Log 1040- Msi install – Event Generated but no atomic indicators
- Application Log 11707- Msi install- “ZeroTier One” in message field
- Application Log 1033- Msi install- “ZeroTier One” in message field
- Application Log 1042- Msi install- Event Generated but no atomic indicators
- Application Log 11724- Msi uninstall- “ZeroTier One” in message field
- Application Log 1034- Msi uninstall- “ZeroTier One” in message field

- System Log 7045- Service install-“ZeroTier One” in message field
- System Log 20001- uPnP install- “zttap300.inf” in message field

- Applications and Services/windows/kernel-pnp Log 400- Device configured- “oem12.inf /zttap/zerotier” in message field
- Applications and Services/windows/kernel-pnp Log 410- Device started- “oem12.inf / zttap” in message field
- Applications and Services/windows/kernel-pnp 420- Device deleted- The class guid in the message field will be the same as the class guids in the 2 logs above (410/400)

- Security Log 5447 rule name "mDNS (UDP-In)"
- Security Log 5447 rule name "mDNS (http-In)"
- Security Log 5156 - Permitted outbound UDP 9993
- Security Log 5158 or 5154 - Bind to local port- Application name is system and source address is not that of the computer (it’s the IP of the ZeroTier Virtual Interface)
- Security Log 4697- A service was installed on a system- “ZeroTier One” present in message field
- Security Log 6419 and 6420 - A request was made to disable a device- “zttap300.inf” present in message field
- Security Log 6421/6422 - A device was enabled- “zttap300.inf” present in message field

Indicators Non-Default Configuration / Behavioral Detections

Event Log Analysis

The following event logs are seen in excess for what appears to be the same device. It wouldn’t be normal to completely build up and tear down the same device or service every x days, hours or minutes. This behavior would indicate that the install happens, it’s used for a short while, then it’s uninstalled.

- System Log 20001- uPnP install
- Applications and Services/windows/kernel-pnp Log 400- Device configured
- Applications and Services/windows/kernel-pnp Log 410- Device started
- Security Log 6421/6422 - A device was enabled
- Applications and Services/windows/kernel-pnp 420- Device deleted
- Security Log 6419 and 6420 - A request was made to disable a device
- Application Log 1040- Msi install
- Application Log 11707- Msi install
- Application Log 1033- Msi install
- Application Log 1042- Msi install
- Application Log 11724- Msi uninstall
- Application Log 1034- Msi uninstall

Registry Analysis

Monitoring the following registry locations for repeated activity from what appears to be the same application. Again, it wouldn't be normal to install and uninstall the same program every x days, hours or minutes. This behavior would indicate that the install happens, it's used for a short while, then it's uninstalled.

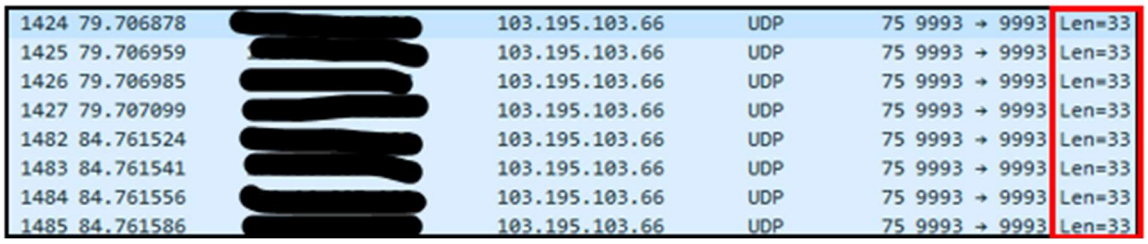
- HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall

Custom Script Example

If feasible, running a custom script that will alert when a new interface is registered on an endpoint and forwarding that as an alert to your SIEM would potentially serve as an indicator of this ZeroTier backdoor. The script would need to obtain a listing of the network interfaces from each device on your network by running an ipconfig, or other equivalent method and check to see if there are any new interfaces.

Network Traffic Analysis

When ZeroTier is installed but the application has yet to join to a network, it was noticed that the endpoint will send UDP packets to the ZeroTier Server that are 33 bytes in length, consistently to the ZeroTier Server from the regular IP interface. Keep in mind that :9993 is the default and this can be changed.



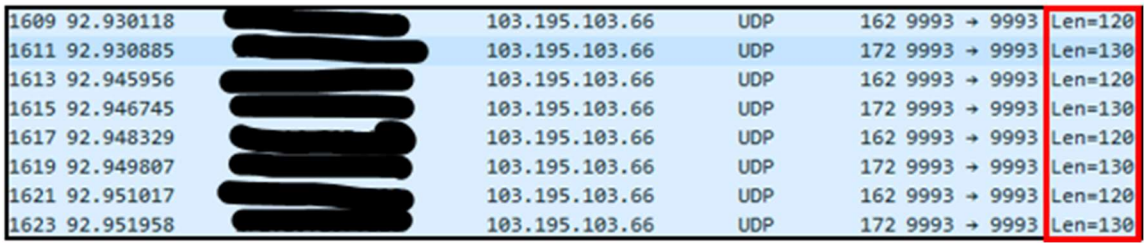
Here is an example Sigma rule to detect the same local IP sending 33 bytes of data to the same remote IP more than 5 times in one minute.

****

```
title: Detect Local IP Sending 33 Bytes of Data to Same Remote IP More Than 5 Times in 1 Minute
status: experimental
description: Detects if ZeroTier is installed but hasn't joined a ZeroTier network yet
logsource:
    category: network_traffic
    product: any
detection:
```

```
    selection:
        bytes_sent: 33
    timeframe: 1m
    condition: selection | count() by src_ip, dest_ip > 5
fields:
    - src_ip
    - dest_ip
    - bytes_sent
falsepositives:
    - Legitimate network traffic with small data packets
level: high
**********
```

Once the endpoint has joined a ZeroTier Network but is sitting idle, in that, nothing interactive is happening, there are some discernable patterns in the packet lengths between the endpoint and the ZeroTier Server. The most reliable pattern is UDP packets with byte lengths that oscillate between 120 and 130. Again, :9993 is the default and can be changed.



Here is an experimental sigma rule that will detect when there is a discernable pattern in the packet lengths from a local IP to the same remote IP. The most reliable pattern is UDP packets with byte lengths that oscillate between 120 and 130.

```
**********

title: Detect Alternating UDP Packet Lengths from Local to Remote IP
status: experimental
description: Detects if ZeroTier is installed and a ZeroTier network has been joined

logsource:
    category: network_traffic
    product: any
detection:
    selection:
        protocol: udp
        lengths:
            - 120
            - 130
    timeframe: 1m
    condition: selection | count() by src_ip, dest_ip, lengths > 5
fields:
    - src_ip
    - dest_ip
    - protocol
    - lengths
falsepositives:
```

- Legitimate network traffic with alternating packet sizes
level: high

****

Conclusion

The default configuration of ZeroTier has more atomic indicators and can be discovered in a straightforward manner if you have the correct event logs, network traffic capture, and access to the endpoints. This configuration would lend itself to laziness on the attacker's part. However, I truly believe that it is reasonable to see this approach in the wild.

If the attacker has taken the time to obfuscate and sanitize the standard metadata, filenames, hashes, etc, discovering it on your network is not so straightforward. It requires the same logs, network traffic capture, and access to the endpoints, but differs in that it would require a programmatic approach through a scripting language to illuminate the patterns required to conclude that the activity is indeed ZeroTier. However, for seasoned analysts, this task is not insurmountable.

Links

Proof of Concept Zero Tier Backdoor YouTube Video Walkthrough
https://www.youtube[.]com/watch?v=0vCblgyvBQo&pp=ygURemVyb3RpZXIgYmFja2Rvb3I%3D

-or just search YouTube for “ZeroTier backdoor”. At the time of writing this, there’s only one YouTube video for this.

Proof of Concept Zero Tier Backdoor Script and Binaries
https://github[.]com/NintendoWii/random/tree/main/ZeroTier

ZeroTier portal
https://zerotier[.]com