Peach Sandstorm deploys new custom Tickler malware in long-running intelligence gathering operations

⋮ 8/28/2024

- By Microsoft Threat Intelligence

Between April and July 2024, Microsoft observed Iranian state-sponsored threat actor Peach Sandstorm deploying a new custom multi-stage backdoor, which we named Tickler. Tickler has been used in attacks against targets in the satellite, communications equipment, oil and gas, as well as federal and state government sectors in the United States and the United Arab Emirates. This activity is consistent with the threat actor's persistent intelligence gathering objectives and represents the latest evolution of their long-standing cyber operations.

Peach Sandstorm also continued conducting password spray attacks against the educational sector for infrastructure procurement and against the satellite, government, and defense sectors as primary targets for intelligence collection. In addition, Microsoft observed intelligence gathering and possible social engineering targeting organizations within the higher education, satellite, and defense sectors via the professional networking platform LinkedIn.

Microsoft assesses that Peach Sandstorm operates on behalf of the Iranian Islamic Revolutionary Guard Corps (IRGC) based on the group's victimology and operational focus. Microsoft further assesses that Peach Sandstorm's operations are designed to facilitate intelligence collection in support of Iranian state interests.

Microsoft tracks Peach Sandstorm campaigns and directly notifies customers who we observe have been targeted or compromised, providing them with the necessary information to help secure their environment. As part of our continuous monitoring, analysis, and reporting on the threat landscape, we are sharing our research on Peach Sandstorm's use of Tickler to raise awareness of this threat actor's evolving tradecraft and to educate organizations on how to harden their attack surfaces against this and similar activity. Microsoft published information on unrelated election interference linked to Iran in the most recent Microsoft Threat Analysis Center (MTAC) report.

Evolution of Peach Sandstorm tradecraft

In past campaigns, Peach Sandstorm has been observed to use password spray attacks to gain access to targets of interest with a high level of success. The threat actor has also conducted intelligence gathering via LinkedIn, researching organizations and individuals employed in the higher education, satellite, and defense sectors.

During the group's latest operations, Microsoft observed new tactics, techniques, and procedures (TTPs) following initial access via password spray attacks or social engineering. Between April and July 2024, Peach Sandstorm deployed a new custom multi-stage backdoor, Tickler, and leveraged Azure infrastructure hosted in fraudulent, attacker-controlled Azure subscriptions for command-and-control (C2). Microsoft continuously monitors Azure, along with all Microsoft products and services, to ensure compliance with our terms of service. Microsoft has notified affected organizations and disrupted the fraudulent Azure infrastructure and accounts associated with this activity.

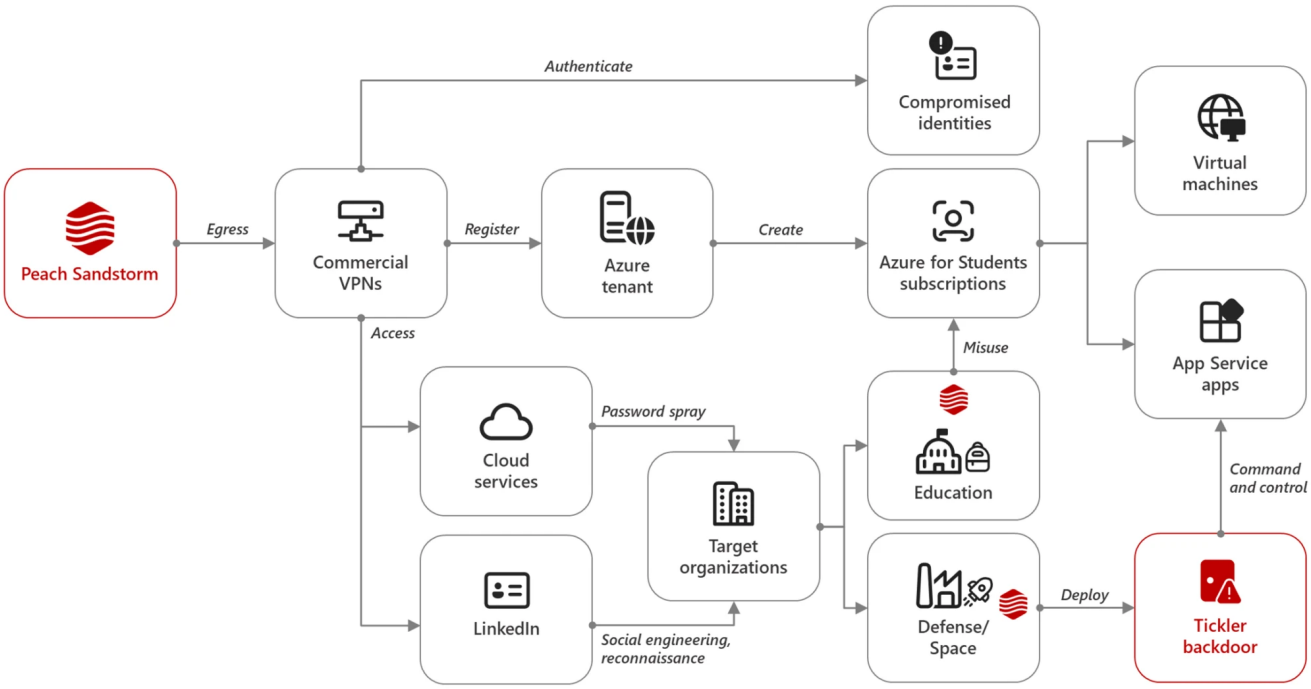

Figure 1. Peach Sandstorm attack chain

Intelligence gathering on LinkedIn

Going back to at least November 2021 and continuing through mid-2024, Microsoft observed Peach Sandstorm using multiple LinkedIn profiles masquerading as students, developers, and talent acquisition managers based in the US and Western Europe. Peach Sandstorm primarily used them to conduct intelligence gathering and possible social engineering against the higher education, satellite sectors, and related industries. The identified LinkedIn accounts were subsequently taken down. Information on LinkedIn's policies and actions against inauthentic behavior on its platform is available here.

Password spray attacks as a common attack vector

Since at least February 2023, Microsoft has observed Peach Sandstorm carrying out password spray activity against thousands of organizations. In password spray attacks, threat actors attempt to authenticate to many different accounts using a single password or a list of commonly used passwords. In contrast to brute force attacks, which target a single account using many passwords, password spray attacks help adversaries maximize their chances for success and minimize the likelihood of automatic account lockouts.

Microsoft has observed that once Peach Sandstorm has verified a target account's credentials using the password spray technique, the threat actor performed subsequent sign-ins to the compromised accounts from commercial VPN infrastructure.

In April and May 2024, Microsoft observed Peach Sandstorm conducting password spray attacks targeting organizations in the defense, space, education, and government sectors in the US and Australia. In particular, Peach Sandstorm continued to use the *"go-http-client"* user agent that they are known to leverage in password spray campaigns. While the password spray activity appeared consistently across sectors, Microsoft observed Peach Sandstorm exclusively leveraging compromised user accounts in the education sector to procure operational infrastructure. In these cases, the threat actor accessed existing Azure subscriptions or created one using the compromised account to host their infrastructure. The attacker-controlled Azure infrastructure then served as C2 or operational hops for Peach Sandstorm operations targeting the government, defense, and space

sectors. Recent updates to security defaults in Azure, such as multi-factor authentication help ensure that Azure accounts are more resistant to account compromise techniques such as those used by Peach Sandstorm.

Tickler malware

Microsoft Threat Intelligence identified two samples of the Tickler malware, a custom multi-stage backdoor, that Peach Sandstorm deployed in compromised environments as recently as July 2024. The first sample was contained in an archive file named *Network Security.zip* alongside benign PDF files used as decoy documents. The archive file contained:

- *YAHSAT NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20240421.pdf.exe* – theTickler malware
- *Yahsat Policy Guide- April 2024.pdf* – a benign PDF
- *YAHSAT NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20240421.pdf* – a second benign PDF

YAHSAT NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20240421.pdf.exe is a 64-bit C/C++ based native PE file. The sample begins with a Process Environment Block (PEB) traversal to locate the in-memory address of file *kernell32.dll*.

Upon successful PEB traversal yielding the address of *kernell32.dll* in memory, the sample decrypts a string to LoadLibraryA and resolves its address, decrypts the string “kernel32.dll”, and loads it again using LoadLibraryA. The sample then launches the benign PDF file *YAHSAT NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20240421.pdf* as a decoy document.

The sample collects the network information from the host and sends it to the C2 URI via HTTP POST request, likely as a means for the threat actor to orient themselves on the compromised network. The below network information is an example generated in a lab environment:

```
Host Name :
DNS Servers :
Node Type : Hybrid
NetBIOS Scope ID :
IP Routing Enabled : no
WINS Proxy Enabled : no
NetBIOS Resolution Uses DNS : noEthernet adapter{                              }
Description : Microsoft Hyper-V Network Adapter #2
Physical Address:
tDHCP Enabled : yes
IP Address : 0.0.0.0
Subnet Mask : 0.0.0.0
Default Gateway : 0.0.0.0
DHCP Server :
Primary WINS Server :
Secondary WINS Server :
Lease Obtained : Wed Dec 31 16:00:00 1969    Lease Expires : Wed Dec 31 16:00:00 1969
```

Figure 2. Network information collected by Tickler after deployment on target host

We subsequently observed Peach Sandstorm iterating and improving on this initial sample. The second Tickler sample, *sold.dll*, is a Trojan dropper functionally identical to the previously identified sample. The malware downloads additional payloads from the C2 server, including a backdoor, a batch script to set persistence for this backdoor, and the following legitimate files:

- *msvcp140.dll* (SHA-256: dad53a78662707d182cdb230e999ef6effc0b259def31c196c51cc3e8c42a9b8)
- *LoggingPlatform.dll* (SHA-256: 56ac00856b19b41bc388ecf749eb4651369e7ced0529e9bf422284070de457b6)

- *vcruntime140.dll* (SHA-256: 22017c9b022e6f2560fee7d544a83ea9e3d85abee367f2f20b3b0448691fe2d4)
- *Microsoft.SharePoint.NativeMessaging.exe* (SHA-256: e984d9085ae1b1b0849199d883d05efbccc92242b1546aeca8afd4b1868c54f5)

The files *msvcp140.dll*, *LoggingPlatform.dll*, *vcruntime140.dll*, and *Microsoft.SharePoint.NativeMessaging.exe* are legitimate Windows signed binaries likely used for DLL sideloading.

Additionally, we observed the sample downloading the following malicious files:

- A batch script (SHA-256: 5df4269998ed79fbc997766303759768ce89ff1412550b35ff32e85db3c1f57b)
- A DLL file (SHA-256: fb70ff49411ce04951895977acfc06fa468e4aa504676dedeb40ba5cea76f37f)
- A DLL file (SHA-256: 711d3deccc22f5acfd3a41b8c8defb111db0f2b474febdc7f20a468f67db0350)

The batch script adds a registry Run key for a file called *SharePoint.exe*, likely used to load the malicious DLL files above, thus setting up persistence:

```
@echo off

reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v
SharePoint /t REG_SZ /d "SharePoint.exe" /f
```

Figure 3. Registry Run key added to set up persistence

The two DLL files are both 64-bit C/C++ compiled PE DLL files and appear to be functionally identical to the previously analyzed samples. As fully functional backdoors, they can run the following commands:

- systeminfo – Gather system information
- dir – List directory
- run – Execute command
- delete – Delete file
- interval – Sleep interval
- upload – Download file from the C2
- download – Upload file to the C2

Azure resources abuse

Microsoft observed Peach Sandstorm creating Azure tenants using Microsoft Outlook email accounts and creating Azure for Students subscriptions in these tenants. Additionally, the group leveraged compromised user accounts in the Azure tenants of organizations in the education sector to do the same. Within these subscriptions, Peach Sandstorm subsequently created Azure resources for use as C2 for the backdoor. Of note, we have observed multiple Iranian groups, including Smoke Sandstorm, use similar techniques in recent months. The following resources were created by Peach Sandstorm for use as Tickler C2 nodes:

- subreviews.azurewebsites[.]net
- satellite2.azurewebsites[.]net
- nodetestservers.azurewebsites[.]net
- satellitegardens.azurewebsites[.]net
- softwareservicesupport.azurewebsites[.]net
- getservicessuports.azurewebsites[.]net
- getservicessupports.azurewebsites[.]net
- getsupportsservices.azurewebsites[.]net
- satellitespecialists.azurewebsites[.]net

- satservicesdev.azurewebsites[.]net
- servicessupports.azurewebsites[.]net
- websupportprotection.azurewebsites[.]net
- supportsoftwarecenter.azurewebsites[.]net
- centersoftwaresupports.azurewebsites[.]net
- softwareservicesupports.azurewebsites[.]net
- getsdervicessupoortss.azurewebsites[.]net

Post-compromise activity

In the past year, Peach Sandstorm has successfully compromised several organizations, primarily in the aforementioned sectors, using bespoke tooling. Once Peach Sandstorm gains access to an organization, the threat actor is known to perform lateral movement and actions on objectives using the following techniques:

Moving laterally via Server Message Block (SMB)

After compromising a European defense organization, Peach Sandstorm threat actors moved laterally via SMB. SMB lateral movement is a technique used by threat actors to move from one compromised machine to another within a network by exploiting the SMB protocol. This protocol, which is used for sharing files, printers, and other resources on a network, could be misused by attackers to propagate their access and gain control over multiple systems.

Downloading and installing a remote monitoring and management (RMM) tool

In an older intrusion against a multinational pharmaceutical company not associated with the campaign discussed in this blog, after a likely successful password spray attack, Peach Sandstorm attempted to download and install AnyDesk, a commercial RMM tool. AnyDesk has a range of capabilities that allow users to remotely access a network, persist in a compromised environment, and enable command and control. The convenience and utility of a tool like AnyDesk is amplified by the fact that it might be permitted by application controls in environments where it is used legitimately by IT support personnel or system administrators.

Taking an Active Directory (AD) snapshot

In at least one intrusion against a Middle East-based satellite operator, Peach Sandstorm actors compromised a user using a malicious ZIP file delivered via Microsoft Teams message followed by dropping AD Explorer and taking an AD snapshot. An AD snapshot is a read-only, point-in-time copy of the AD database and related files, which can be used for various legitimate administrative tasks. These snapshots can also be exploited by threat actors for malicious purposes.

Mitigations

To harden networks against Peach Sandstorm activity, defenders can implement the following:

- Reset account passwords for any accounts targeted during a password spray attack. If a targeted account had system-level permissions, further investigation may be warranted.
- Revoke session cookies in addition to resetting passwords.
    - Revoke any MFA setting changes made by the attacker on any compromised users' accounts.
    - Require re-challenging MFA for MFA updates as the default.
- Implement the Azure Security Benchmark and general best practices for securing identity infrastructure, including:

    - Create conditional access policies to allow or disallow access to the environment based on defined criteria.
    - Block legacy authentication with Microsoft Entra by using Conditional Access. Legacy authentication protocols don't have the ability to enforce multifactor authentication (MFA), so blocking such authentication methods will help prevent password spray attackers from taking advantage of the lack of MFA on those protocols.
    - Enable AD FS web application proxy extranet lockout to protect users from potential password brute force compromise.
- Secure accounts with credential hygiene:
    - Practice the principle of least privilege and audit privileged account activity in your Microsoft Entra environments to help slow and stop attackers.
    - Deploy Microsoft Entra Connect Health for Active Directory Federation Services (AD FS). This captures failed attempts as well as IP addresses recorded in AD FS logs for bad requests in the *Risky IP report*.
    - Use Microsoft Entra password protection to help detect and block known weak passwords and their variants.
    - Turn on identity protection in Microsoft Entra to monitor for identity-based risks and create policies for risky sign ins.
- Comply with the recent MFA enforcement policy requiring all Azure accounts to utilize MFA. Keep MFA always-on for privileged accounts and apply risk-based MFA for normal accounts.
    - Consider transitioning to a passwordless primary authentication method, such as Azure MFA, certificates, or Windows Hello for Business.
- Secure remote desktop protocol (RDP) or Windows Virtual Desktop endpoints with MFA to harden against password spray or brute force attacks.

To protect against password spray attacks, implement the following mitigations:

- Eliminate insecure passwords.
- Educate users to review sign-in activity and mark suspicious sign-in attempts as "This wasn't me".
- Reset account passwords for any accounts targeted during a password spray attack. If a targeted account had system-level permissions, further investigation may be warranted.
- Detect, investigate, and remediate identity-based attacks using solutions like Microsoft Entra ID Protection.
- Investigate compromised accounts using Microsoft Purview Audit (Premium).
- Enforce on-premises Microsoft Entra Password Protection for Microsoft Active Directory Domain Services.
- Use risk detections for user sign-ins to trigger multifactor authentication or password changes.
- Investigate any possible password spray activity using the password spray investigation playbook.

Strengthen endpoints against attacks by following these steps:

- Turn on cloud-delivered protection in Microsoft Defender Antivirus or the equivalent for your antivirus product to help cover rapidly evolving attacker tools and techniques.
- Enable real-time protection in Microsoft Defender Antivirus or the equivalent for your antivirus product.
- Detect and block potentially unwanted applications through Microsoft Defender for Endpoint.
- Run endpoint detection and response (EDR) in block mode so that Microsoft Defender for Endpoint can help block malicious artifacts, even when your non-Microsoft antivirus does not detect the threat, or when Microsoft Defender Antivirus is running in passive mode. EDR in block mode works behind the scenes to help remediate malicious artifacts that are detected post-compromise.
- Turn on attack surface reduction rules to help prevent common attack techniques:
    - Block executable files from running unless they meet a prevalence, age, or trusted list criterion

    - Block execution of potentially obfuscated scripts
- Implement anomaly detection policies in Microsoft Defender for Cloud Apps.
- Enable protections in Microsoft Defender for Endpoint to help safeguard against malicious sites and internet-based threats.
    - Network protection
    - Web protection
- Enable tamper protection within Microsoft Defender for Endpoint to help prevent threat actors from disabling or changing security features, such as virus and threat protection.

Microsoft Defender XDR detections

Microsoft Defender Antivirus

Microsoft Defender Antivirus detects components of this threat as the following malware:

- TrojanDownloader:Win64/Tickler
- Backdoor:Win64/Tickler

Microsoft Defender for Endpoint

The following Microsoft Defender for Endpoint alerts can indicate associated threat activity:

- Peach Sandstorm actor activity detected

The following alerts might also indicate threat activity related to this threat. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Password spraying
- Unfamiliar Sign-in properties
- An executable file loaded an unexpected DLL file

Microsoft Defender for Identity

The following Microsoft Defender for Identity alerts can indicate activity related to this threat. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Atypical travel
- Suspicious behavior: Impossible travel activity

Microsoft Defender for Cloud Apps

The following Microsoft Defender for Cloud Apps alerts can indicate activity related to this threat. Note, however, that these alerts can be also triggered by unrelated threat activity.

- Activity from a Tor IP address
- Suspicious Administrative Activity
- Impossible travel activity
- Multiple failed login attempts
- Activity from an anonymous proxy

Threat intelligence reports

Microsoft Defender Threat Intelligence customers can use the following reports in Microsoft products to get the most up-to-date information about the threat actor, malicious activity, and techniques discussed in this blog. These reports provide the intelligence, protection information, and recommended actions to help prevent, mitigate, or respond to associated threats found in customer environments.

Microsoft Defender Threat Intelligence

Hunting queries

Microsoft Defender XDR

Microsoft Defender XDR customers can run the following query to find related activity in their networks:

The following query identifies failed attempts to sign-in from multiple sources that originate from a single ISP. Attackers distribute attacks from multiple IP addresses across a single service provider to evade detection. Run query

```
IdentityLogonEvents

| where Timestamp > ago(4h)

| where ActionType == "LogonFailed"

| where isnotempty(AccountObjectId)

| summarize TargetCount = dcount(AccountObjectId), TargetCountry = dcount(Location),
TargetIPAddress = dcount(IPAddress) by ISP

| where TargetCount >= 100

| where TargetCountry >= 5

| where TargetIPAddress >= 25
```

Connectivity to C2s

The following queries identifies connectivity to Peach Sandstorm created Azure App Service apps for command and control. Run query

```
let domainList = dynamic(["subreviews.azurewebsites.net",

"satellite2.azurewebsites.net",

"nodetestservers.azurewebsites.net",

"satellitegardens.azurewebsites.net",

"softwareservicesupport.azurewebsites.net",

"getservicessuports.azurewebsites.net",

"getservicessupports.azurewebsites.net",

"getsupportsservices.azurewebsites.net",

"satellitespecialists.azurewebsites.net",

"satservicesdev.azurewebsites.net",

"servicessupports.azurewebsites.net",

"websupportprotection.azurewebsites.net ",
```

```
"supportsoftwarecenter.azurewebsites.net",

"centersoftwaresupports.azurewebsites.net"

"softwareservicesupports.azurewebsites.net",

"getsdervicessupoortss.azurewebsites.net"]);union

(

DnsEvents

| where QueryType has_any(domainList) or Name has_any(domainList)

| project TimeGenerated, Domain = QueryType, SourceTable = "DnsEvents"

),

(

IdentityQueryEvents

| where QueryTarget has_any(domainList)

| project Timestamp, Domain = QueryTarget, SourceTable = "IdentityQueryEvents"

),

(

DeviceNetworkEvents

| where RemoteUrl has_any(domainList)

| project Timestamp, Domain = RemoteUrl, SourceTable = "DeviceNetworkEvents"

),

(

DeviceNetworkInfo

| extend DnsAddresses = parse_json(DnsAddresses), ConnectedNetworks =
parse_json(ConnectedNetworks)

| mv-expand DnsAddresses, ConnectedNetworks

| where DnsAddresses has_any(domainList) or ConnectedNetworks.Name has_any(domainList)

| project Timestamp, Domain = coalesce(DnsAddresses, ConnectedNetworks.Name), SourceTable =
"DeviceNetworkInfo"

),

(

VMConnection

| extend RemoteDnsQuestions = parse_json(RemoteDnsQuestions), RemoteDnsCanonicalNames =
parse_json(RemoteDnsCanonicalNames)

| mv-expand RemoteDnsQuestions, RemoteDnsCanonicalNames

| where RemoteDnsQuestions has_any(domainList) or RemoteDnsCanonicalNames has_any(domainList)

| project TimeGenerated, Domain = coalesce(RemoteDnsQuestions, RemoteDnsCanonicalNames),
SourceTable = "VMConnection"

),
```

```
(
W3CIISLog
| where csHost has_any(domainList) or csReferer has_any(domainList)
| project TimeGenerated, Domain = coalesce(csHost, csReferer), SourceTable = "W3CIISLog"
),
(
EmailUrlInfo
| where UrlDomain has_any(domainList)
| project Timestamp, Domain = UrlDomain, SourceTable = "EmailUrlInfo"
),
(
UrlClickEvents
| where Url has_any(domainList)
| project Timestamp, Domain = Url, SourceTable = "UrlClickEvents"
)
| order by TimeGenerated desc
```

Malicious file activity

The following query will surface events involving malicious files related to this activity. Run query

```
let fileHashes = dynamic(["711d3deccc22f5acfd3a41b8c8defb111db0f2b474febdc7f20a468f67db0350",
"fb70ff49411ce04951895977acfc06fa468e4aa504676dedeb40ba5cea76f37f",
"5df4269998ed79fbc997766303759768ce89ff1412550b35ff32e85db3c1f57b",
"ccb617cc7418a3b22179e00d21db26754666979b4c4f34c7fda8c0082d08cec4",
"7eb2e9e8cd450fc353323fd2e8b84fbbdfe061a8441fd71750250752c577d198"]);
union
(
DeviceFileEvents
| where SHA256 in (fileHashes)
| project Timestamp, FileHash = SHA256, SourceTable = "DeviceFileEvents"
),
(
DeviceEvents
| where SHA256 in (fileHashes)
| project Timestamp, FileHash = SHA256, SourceTable = "DeviceEvents"
),
(
DeviceImageLoadEvents
```

```
| where SHA256 in (fileHashes)
| project Timestamp, FileHash = SHA256, SourceTable = "DeviceImageLoadEvents"
),
(
DeviceProcessEvents
| where SHA256 in (fileHashes)
| project Timestamp, FileHash = SHA256, SourceTable = "DeviceProcessEvents"
)
| order by Timestamp desc
```

Microsoft Sentinel

Microsoft Sentinel customers can use the TI Mapping analytics (a series of analytics all prefixed with 'TI map') to automatically match the malicious domain indicators mentioned in this blog post with data in their workspace. If the TI Map analytics are not currently deployed, customers can install the Threat Intelligence solution from the Microsoft Sentinel Content Hub to have the analytics rule deployed in their Sentinel workspace.

Indicators of compromise

Domains

- subreviews.azurewebsites[.]net
- satellite2.azurewebsites[.]net
- nodetestservers.azurewebsites[.]net
- satellitegardens.azurewebsites[.]net
- softwareservicesupport.azurewebsites[.]net
- getservicessuports.azurewebsites[.]net
- getservicessupports.azurewebsites[.]net
- getsupportsservices.azurewebsites[.]net
- satellitespecialists.azurewebsites[.]net
- satservicesdev.azurewebsites[.]net
- servicessupports.azurewebsites[.]net
- websupportprotection.azurewebsites[.]net
- supportsoftwarecenter.azurewebsites[.]net
- centersoftwaresupports.azurewebsites[.]net
- softwareservicesupports.azurewebsites[.]net
- getsdervicessupoortss.azurewebsites[.]net

Tickler samples and related indicators

- *YAHSAT NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20240421.pdf.exe* (SHA-256: 7eb2e9e8cd450fc353323fd2e8b84fbbdfe061a8441fd71750250752c577d198)
- *Sold.dll* (SHA-256: ccb617cc7418a3b22179e00d21db26754666979b4c4f34c7fda8c0082d08cec4)
- Batch script (SHA-256: 5df4269998ed79fbc997766303759768ce89ff1412550b35ff32e85db3c1f57b)
- Malicious DLL (SHA-256: fb70ff49411ce04951895977acfc06fa468e4aa504676dedeb40ba5cea76f37f)
- Malicious DLL (SHA-256: 711d3deccc22f5acfd3a41b8c8defb111db0f2b474febdc7f20a468f67db0350)