Src: Analysis of Storm-0558 techniques for unauthorized email access | Microsoft Security Blog

Analysis of Storm-0558 techniques for unauthorized email access

By Microsoft Threat Intelligence

July 14, 2023

Threat actors

Executive summary

On July 11, 2023, Microsoft published two blogs detailing a malicious campaign by a threat actor tracked as Storm-0558 that targeted customer email that we've detected and mitigated: Microsoft Security Response Center and Microsoft on the Issues. As we continue our investigation into this incident and deploy defense in depth measures to harden all systems involved, we're providing this deeper analysis of the observed actor techniques for obtaining unauthorized access to email data, tools, and unique infrastructure characteristics.

As described in more detail in our July 11 blogs, Storm-0558 is a China-based threat actor with espionage objectives. Beginning May 15, 2023, Storm-0558 used forged authentication tokens to access user email from approximately 25 organizations, including government agencies and related consumer accounts in the public cloud. No other environment was impacted. Microsoft has successfully blocked this campaign from Storm-0558. As with any observed nation-state actor activity, Microsoft has directly notified targeted or compromised customers, providing them with important information needed to secure their environments.

Since identification of this malicious campaign on June 16, 2023, Microsoft has identified the root cause, established durable tracking of the campaign, disrupted malicious activities, hardened the environment, notified every impacted customer, and coordinated with multiple government entities. We continue to investigate and monitor the situation and will take additional steps to protect customers.

Actor overview

Microsoft Threat Intelligence assesses with moderate confidence that Storm-0558 is a China-based threat actor with activities and methods consistent with espionage objectives. While we have discovered some minimal overlaps with other Chinese groups such as Violet Typhoon (ZIRCONIUM, APT31), we maintain high confidence that Storm-0558 operates as its own distinct group.

Figure 1 shows Storm-0558 working patterns from April to July 2023; the actor's core working hours are consistent with working hours in China, Monday through Friday from

Commented [1]: What was forged? (Timestamp, issuer)
How was the forgery conducted?
How did the verification system fail to detect the forgery?

Commented [2]: How can you be sure? This statement is not as reassuring as the author intended.

Commented [3]: What does block mean here? Blocked how? What did you block?

The statement is way too definitive.
- Either the author doesn't know the full-scope, or is lying.
Misleading statement at best. Redundant posit.

Commented [4]: What do these words mean? Disrupt what?
Did MSFT send a AIM-9 Siwdewinder on a building, or a AGM114-R9x on a car?

12:00 AM UTC (8:00 AM China Standard time) through 09:00 AM UTC (5:00 PM China Standard Time).

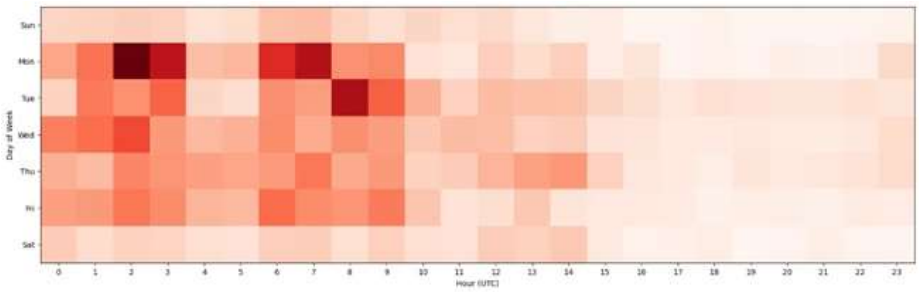


Figure 1. Heatmap of observed Stom-0558 activity by day of week and hour (UTC).

In past activity observed by Microsoft, Storm-0558 has primarily targeted US and European diplomatic, economic, and legislative governing bodies, and individuals connected to Taiwan and Uyghur geopolitical interests.

Historically, this threat actor has displayed an interest in targeting media companies, think tanks, and telecommunications equipment and service providers. The objective of most Storm-0558 campaigns is to obtain unauthorized access to email accounts belonging to employees of targeted organizations. Storm-0558 pursues this objective through credential harvesting, phishing campaigns, and OAuth token attacks. This threat actor has displayed an interest in OAuth applications, token theft, and token replay against Microsoft accounts since at least August 2021. Storm-0558 operates with a high degree of technical tradecraft and operational security. The actors are keenly aware of the target's environment, logging policies, authentication requirements, policies, and procedures. Storm-0558's tooling and reconnaissance activity suggests the actor is technically adept, well resourced, and has an in-depth understanding of many authentication techniques and applications.

In the past, Microsoft has observed Storm-0558 obtain credentials for initial access through phishing campaigns. The actor has also exploited vulnerabilities in public-facing applications to gain initial access to victim networks. These exploits typically result in web shells, including *China Chopper*, being deployed on compromised servers. One of the most prevalent malware families used by Storm-0558 is a shared tool tracked by Microsoft as *Cigril*. This family exists in several variants and is launched using dynamic-link library (DLL) search order hijacking.

After gaining access to a compromised system, Storm-0558 accesses credentials from a variety of sources, including the LSASS process memory and Security Account Manager (SAM) registry hive. Microsoft assesses that once Storm-0558 has access to the desired user credentials, the actor signs into the compromised user's cloud email account with the valid account credentials. The actor then collects information from the email account over the web service.

Initial discovery and analysis of current activity

Commented [5]: The bane of our existence.

Commented [6]: Wait, is this an on-prem to Cloud pivot?
What is the compromised system?

Commented [7]: Are we still on the MailItemsAccessed story, is this a different story? This is really confusing.

OR is there an effort to paint different things that Storm-0558 does, and in that case -- please include their prowess in high-jump as well.

Please don't mix stories.

On June 16, 2023, Microsoft was notified by a customer of anomalous Exchange Online data access. Microsoft analysis attributed the activity to Storm-0558 based on established prior TTPs. We determined that Storm-0558 was accessing the customer's Exchange Online data using Outlook Web Access (OWA). Microsoft's investigative workflow initially assumed the actor was stealing correctly issued Azure Active Directory (Azure AD) tokens, most probably using malware on infected customer devices. **Microsoft analysts later determined that the actor's access was utilizing Exchange Online authentication artifacts, which are typically derived from Azure AD authentication tokens (Azure AD tokens)**. Further in-depth analysis over the next several days led Microsoft analysts to assess that the internal Exchange Online authentication artifacts did not correspond to Azure AD tokens in Microsoft logs.

Microsoft analysts began investigating the possibility that the actor was forging authentication tokens using an acquired Azure AD enterprise signing key. **In-depth analysis of the Exchange Online activity discovered that in fact the actor was forging Azure AD tokens using an acquired Microsoft account (MSA) consumer signing key**. This was made possible by a validation error in Microsoft code. The use of an incorrect key to sign the requests allowed our investigation teams to see all actor access requests which followed this pattern across both our enterprise and consumer systems. Use of the incorrect key to sign this scope of assertions was an obvious indicator of the actor activity as no Microsoft system signs tokens in this way. Use of acquired signing material to forge authentication tokens to access customer Exchange Online data differs from previously observed Storm-0558 activity. Microsoft's investigations have not detected any other use of this pattern by other actors and Microsoft has taken steps to block related abuse.

Actor techniques

Token forgery

Authentication tokens are used to validate the identity of entities requesting access to resources – in this case, email. These tokens are issued to the requesting entity (such as a user's browser) by identity providers like Azure AD. To prove authenticity, the identity provider signs the token using a private signing key. The relying party validates the token presented by the requesting entity by using a public validation key. Any request whose signature is correctly validated by the published public validation key will be trusted by the relying party. An actor that can acquire a private signing key can then create falsified tokens with valid signatures that will be accepted by relying parties. This is called token forgery.

Storm-0558 acquired an inactive MSA consumer signing key and used it to forge authentication tokens for Azure AD enterprise and MSA consumer to access OWA and Outlook.com. All MSA keys active prior to the incident – including the actor-acquired MSA signing key – have been invalidated. Azure AD keys were not impacted. The method by which the actor acquired the key is a matter of ongoing investigation. Though the key was intended only for MSA accounts, a validation issue allowed this key to be trusted for signing Azure AD tokens. This issue has been corrected.

Commented [8]: *Accessing from where?*
I am assuming the VPN/Tor details below is where this was accessed from? Please confirm.

Commented [9]: This is a bombshell disclosure.

- How do you access Exchange artefacts? Using your on-prem AD, or AAD tokens right? RIGHT ??
- Is there a SEPARATE Auth mechanism for accessing Exchange artefacts that DOES-NOT include AD/AAD ?

In that case, that would constitute a backdoor in EXO :|

Commented [10]: This is WILD.
How can you access Exchange calendar/notes/mail without a corresponding AD/AAD token?

Are you sure there is no backdoor here, or whatever we learnt in last 25 years was all wrong ?

Commented [11]: *This is the heart of the issue.*
Need further details about what was the validation error, Cipher-suite/date checking/cert-chain propagation/bad-defaults / something else.
Also trying to understand the consequence of validation error, and if this is limited to OWA only, or does it extend to other products.
VVV Important finding.

Commented [12]: I understand this is an observation from DFIR - but what does it mean in technical terms. Is scope of the assertion - ReadAllEmails ? Please be specific.

Commented [13]: *Novelty factor* in the attack.
Waiting on Orange Tsai/Dirk-Jaan to comment on this -- but details are sparse.

Commented [14]: Like a backup API key?
How would MSFT lose a backup key, unless that key was used for more than 1-purpose?
Both primary/secondary keys would be logged in the same KMS?
Investigate further

Commented [15]: Bravo. Correct response.
Reissue all keys

Commented [16]: A-HA !!
Dual-use of key confirmed. DING-DING-DING.

So, MSFT is *still looking for initial intrusion vector?*

As part of defense in depth, we continuously update our systems. We have substantially hardened key issuance systems since the acquired MSA key was initially issued. This includes increased isolation of the systems, refined monitoring of system activity, and moving to the hardened key store used for our enterprise systems. We have revoked all previously active keys and issued new keys using these updated systems. Our active investigation indicates these hardening and isolation improvements disrupt the mechanisms we believe the actor could have used to acquire MSA signing keys. No key-related actor activity has been observed since Microsoft invalidated the actor-acquired MSA signing key. Further, we have seen Storm-0558 transition to other techniques, which indicates that the actor is not able to utilize or access any signing keys. We continue to explore other ways the key may have been acquired and add additional defense in depth measures.

Commented [17]: I thought you guys already had that. Do we need to revisit MSFT's own hardening systems etc. ?

Commented [18]: Ok -- But if you don't know how you got there, what you have done here is segregated all systems and ensured there are no tight coupling / dual-use keys.
Since you don't know how they got in -- you can't fix that part -YET.

Identity techniques for access

Once authenticated through a legitimate client flow leveraging the forged token, the threat actor accessed the OWA API to retrieve a token for Exchange Online from the GetAccessTokenForResource API used by OWA. The actor was able to obtain new access tokens by presenting one previously issued from this API due to a design flaw. This flaw in the GetAccessTokenForResourceAPI has since been fixed to only accept tokens issued from Azure AD or MSA respectively. The actor used these tokens to retrieve mail messages from the OWA API.

Commented [19]: 2nd important observation, and one of the most *common logic bugs.*
Issue a new token, if presented with a previous expired token (Eng would code this in for safety, so that they don't lockout consumers from systems.)

So someone grabs the 3:00pm ticket stub in the parking lot of a Cinema, and presents it at gate for the 6:00pm show.
The usher lets them in.

Actor tooling

Microsoft Threat Intelligence routinely identifies threat actor capabilities and leverages file intelligence to facilitate our protection of Microsoft customers. During this investigation, we identified several distinct Storm-0558 capabilities that facilitate the threat actor's intrusion techniques. The capabilities described in this section are not expected to be present in the victim environment.

Storm-0558 uses a collection of PowerShell and Python scripts to perform REST API calls against the OWA Exchange Store service. For example, Storm-0558 has the capability to use minted access tokens to extract email data such as:

- Download emails
- Download attachments
- Locate and download conversations
- Get email folder information

The generated web requests can be routed through a Tor proxy or several hardcoded SOCKS5 proxy servers. The threat actor was observed using several User-Agents when issuing web requests, for example:

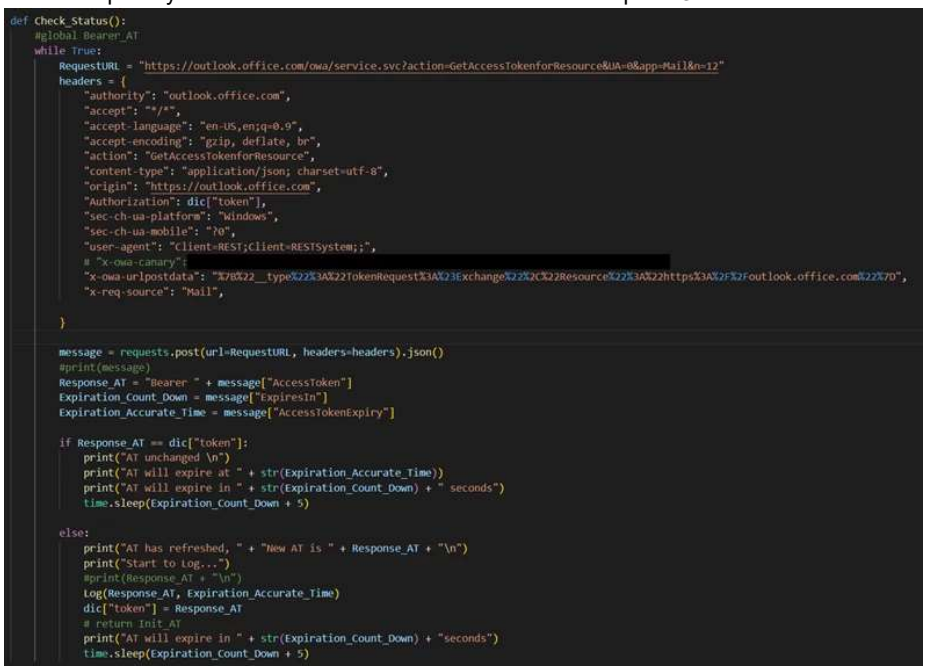
- Client=REST;Client=RESTSystem;;
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36

- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
- “Microsoft Edge”;v=”113″, “Chromium”;v=”113″, “Not-A.Brand”;v=”24″

The scripts contain highly sensitive hardcoded information such as bearer access tokens and email data, which the threat actor uses to perform the OWA API calls. The threat actor has the capability to refresh the access token for use in subsequent OWA commands.

Commented [20]: This is what MSFT managed to catch.
There is *still no information on how they obtained the bearer tokens in the first-place.*

Once you have a bearer token, rest of the stuff is just stealth & opsec.



```
def Check_Status():
    #global Bearer_AT
    while True:
        RequestURL = "https://outlook.office.com/owa/service.svc?action=GetAccessTokenforResource&UA=0&app=Mail&n=12"
        headers = {
            "authority": "outlook.office.com",
            "accept": "*/*",
            "accept-language": "en-US,en;q=0.9",
            "accept-encoding": "gzip, deflate, br",
            "action": "GetAccessTokenforResource",
            "content-type": "application/json; charset=utf-8",
            "origin": "https://outlook.office.com",
            "Authorization": dic["token"],
            "sec-ch-ua-platform": "Windows",
            "sec-ch-ua-mobile": "?0",
            "user-agent": "Client=REST;Client=RESTSystem;;",
            # "x-owa-canary":
            "x-owa-urlpostdata": "%7B%22__type%22%3A%22TokenRequest%3A%23Exchange%22%2C%22Resource%22%3A%22https%3A%2F%2Foutlook.office.com%22%7D",
            "x-req-source": "Mail",

        }

        message = requests.post(url=RequestURL, headers=headers).json()
        #print(message)
        Response_AT = "Bearer " + message["AccessToken"]
        Expiration_Count_Down = message["ExpiresIn"]
        Expiration_Accurate_Time = message["AccessTokenExpiry"]

        if Response_AT == dic["token"]:
            print("AT unchanged \n")
            print("AT will expire at " + str(Expiration_Accurate_Time))
            print("AT will expire in " + str(Expiration_Count_Down) + " seconds")
            time.sleep(Expiration_Count_Down + 5)

        else:
            print("AT has refreshed, " + "New AT is " + Response_AT + "\n")
            print("Start to Log...")
            #print(Response_AT + "\n")
            Log(Response_AT, Expiration_Accurate_Time)
            dic["token"] = Response_AT
            # return Init_AT
            print("AT will expire in " + str(Expiration_Count_Down) + "seconds")
            time.sleep(Expiration_Count_Down + 5)
```

Figure 2. Python code snippet of the token refresh functionality used by the threat actor.

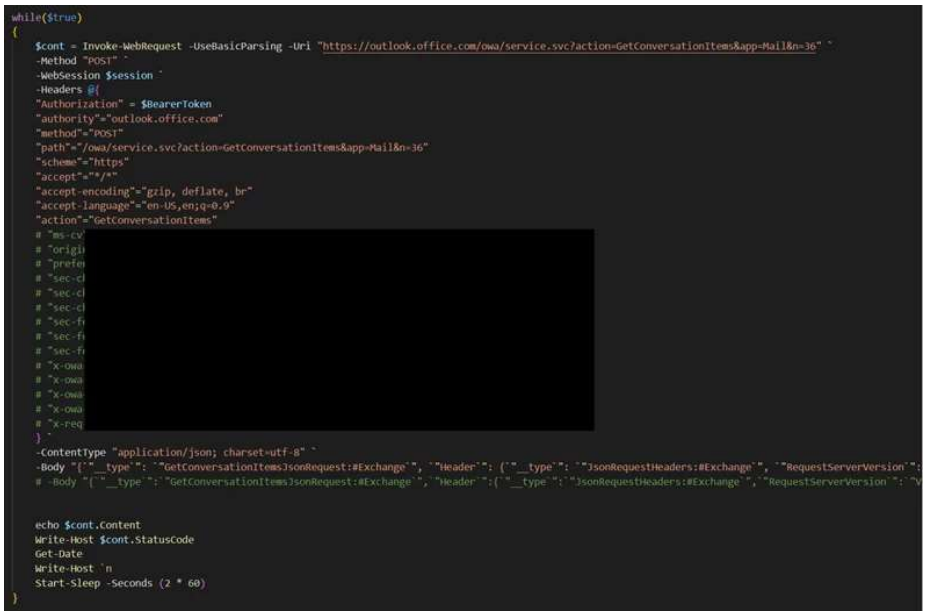


Figure 3. PowerShell code snippet of OWA REST API call to GetConversationItems.

Actor infrastructure

During significant portions of Storm-0558's malicious activities, the threat actor leveraged dedicated infrastructure running the SoftEther proxy software. Proxy infrastructure complicates detection and attribution of Storm-0558 activities. During our response, Microsoft Threat Intelligence identified a unique method of profiling this proxy infrastructure and correlated with behavioral characteristics of the actor intrusion techniques. Our profile was based on the following facets:

1. Hosts operating as part of this network present a JARM fingerprint consistent with SoftEther VPN: 06d06d07d06d06d06c42d42d000000cdb95e27fd8f9fee4a2bec829b889b8b.
2. Presented x509 certificate has expiration date of December 31, 2037.
3. Subject information within the x509 certificate does not contain "softether".

Over the course of the campaign, the IPs listed in the table below were used during the corresponding timeframes.

IP address	First seen	Last seen	Description

51.89.156[.]153	3/9/2023	7/10/2023	SoftEther proxy
176.31.90[.]129	3/28/2023	6/29/2023	SoftEther proxy
137.74.181[.]100	3/31/2023	7/11/2023	SoftEther proxy
193.36.119[.]45	4/19/2023	7/7/2023	SoftEther proxy
185.158.248[.]159	4/24/2023	7/6/2023	SoftEther proxy
131.153.78[.]188	5/6/2023	6/29/2023	SoftEther proxy
37.143.130[.]146	5/12/2023	5/19/2023	SoftEther proxy
146.70.157[.]45	5/12/2023	6/8/2023	SoftEther proxy
185.195.200[.]39	5/15/2023	6/29/2023	SoftEther proxy

185.38.142[.]229	5/15/2023	7/12/2023	SoftEther proxy
146.70.121[.]44	5/17/2023	6/29/2023	SoftEther proxy
31.42.177[.]181	5/22/2023	5/23/2023	SoftEther proxy
185.51.134[.]52	6/7/2023	7/11/2023	SoftEther proxy
173.44.226[.]70	6/9/2023	7/11/2023	SoftEther proxy
45.14.227[.]233	6/12/2023	6/26/2023	SoftEther proxy
185.236.231[.]109	6/12/2023	7/3/2023	SoftEther proxy
178.73.220[.]149	6/16/2023	7/12/2023	SoftEther proxy
45.14.227[.]212	6/19/2023	6/29/2023	SoftEther proxy

91.222.173[.]225	6/20/2023	7/1/2023	SoftEther proxy
146.70.35[.]168	6/22/2023	6/29/2023	SoftEther proxy
146.70.157[.]213	6/26/2023	6/30/2023	SoftEther proxy
31.42.177[.]201	6/27/2023	6/29/2023	SoftEther proxy
5.252.176[.]8	7/1/2023	7/1/2023	SoftEther proxy
80.85.158[.]215	7/1/2023	7/9/2023	SoftEther proxy
193.149.129[.]88	7/2/2023	7/12/2023	SoftEther proxy
5.252.178[.]68	7/3/2023	7/11/2023	SoftEther proxy
116.202.251[.]8	7/4/2023	7/7/2023	SoftEther proxy

185.158.248[.]93	6/25/2023	06/26/2023	SoftEther proxy
20.108.240[.]252	6/25/2023	7/5/2023	SoftEther proxy
146.70.135[.]182	5/18/2023	6/22/2023	SoftEther proxy

As early as May 15, 2023, Storm-0558 shifted to using a separate series of dedicated infrastructure servers specifically for token replay and interaction with Microsoft services. It is likely that the dedicated infrastructure and supporting services configured on this infrastructure offered a more efficient manner of facilitating the actor's activities. The dedicated infrastructure would host an actor-developed web panel that presented an authentication page at URI */#/login*. The observed sign-in pages had one of two SHA-1 hashes: 80d315c21fc13365bba5b4d56357136e84ecb2d4 and 931e27b6f1a99edb96860f840eb7ef201f6c68ec.

Commented [21]: Great job, but let's not extend the file-based Windows attack mental models to web-based attacks.

The idea of tracking SHA-1s for every visited webpage should not extend beyond proxy servers. Specially not in DFIR/Infosec space

Figure 4. Token web panel sign-in page with SHA-1 hashes.

As part of the intelligence-driven response to this campaign, and in support of tracking, analyzing, and disrupting actor activity, analytics were developed to proactively track the dedicated infrastructure. Through this tracking, we identified the following dedicated infrastructure.

IP address	First seen	Last seen	Description
195.26.87[.]219	5/15/2023	6/25/2023	Token web panel
185.236.228[.]183	5/24/2023	6/11/2023	Token web panel
85.239.63[.]160	6/7/2023	6/11/2023	Token web panel
193.105.134[.]58	6/24/2023	6/25/2023	Token web panel
146.0.74[.]16	6/28/2023	7/4/2023	Token web panel
91.231.186[.]226	6/29/2023	7/4/2023	Token web panel
91.222.174[.]41	6/29/2023	7/3/2023	Token web panel
185.38.142[.]249	6/29/2023	7/2/2023	Token web panel

The last observed dedicated token replay infrastructure associated with this activity was stood down on July 4, 2023, roughly one day following the coordinated mitigation conducted by Microsoft.

Post-compromise activity

Our telemetry and investigations indicate that post-compromise activity was limited to email access and exfiltration for targeted users.

Mitigation and hardening

No customer action is required to mitigate the token forgery technique or validation error in OWA or Outlook.com. Microsoft has mitigated this issue on customers' behalf as follows:

- On June 26, OWA stopped accepting tokens issued from GetAccessTokensForResource for renewal, which mitigated the token renewal being abused.
- On June 27, Microsoft blocked the usage of tokens signed with the acquired MSA key in OWA preventing further threat actor enterprise mail activity.
- On June 29, Microsoft completed replacement of the key to prevent the threat actor from using it to forge tokens. Microsoft revoked all MSA signing which were valid at the time of the incident, including the actor-acquired MSA key. The new

Commented [22]: This is hard to do in PROD, specially for someone as large as MSFT. This is a prod API change. Not sure how they could do it without breaking legitimate customer access.
Kudos.

Commented [23]: Again - doing this kinda stuff at scale for XX millions of users without causing an outage -- HATS OFF !!
Pulling this off is an achievement in itself.
Doing it under pressure and in the middle of an incident within 2 days - Kudos.

- MSA signing keys are issued in substantially updated systems

Commented [24]: What does this mean ??

- which benefit from hardening not present at issuance of the actor-acquired MSA key:
    - Microsoft has increased the isolation of these systems from corporate environments, applications, and users.Microsoft has refined monitoring of all systems related to key activity, and increased automated alerting related to this monitoring.
    - Microsoft has moved the MSA signing keys to the key store used for our enterprise systems.
- On July 3, Microsoft blocked usage of the key for all impacted consumer customers to prevent use of previously-issued tokens.

Commented [25]: HSM or Azure keyvault or something else?

Ongoing monitoring indicates that all actor activity related to this incident has been blocked. Microsoft will continue to monitor Storm-0558 activity and implement protections for our customers.

Recommendations

Microsoft has mitigated this activity on our customers' behalf for Microsoft services. No customer action is required to prevent threat actors from using the techniques described above to access Exchange Online and Outlook.com.

Indicators of compromise

Indicator	Type	First seen	Last seen	Description
d4b4cccda9228624656bff33d8110955779632aa	Thumbprint			Thumbprint of acquired signing key
195.26.87[.]219	IPv4	5/15/2023	6/25/2023	Token web panel
185.236.228[.]183	IPv4	5/24/2023	6/11/2023	Token web panel

85.239.63[.]160	IPv4	6/7/2023	6/11/2023	Token web panel
193.105.134[.]58	IPv4	6/24/2023	6/25/2023	Token web panel
146.0.74[.]16	IPv4	6/28/2023	7/4/2023	Token web panel
91.231.186[.]226	IPv4	6/29/2023	7/4/2023	Token web panel
91.222.174[.]41	IPv4	6/29/2023	7/3/2023	Token web panel
185.38.142[.]249	IPv4	6/29/2023	7/2/2023	Token web panel
51.89.156[.]153	IPv4	3/9/2023	7/10/2023	SoftEther proxy
176.31.90[.]129	IPv4	3/28/2023	6/29/2023	SoftEther proxy
137.74.181[.]100	IPv4	3/31/2023	7/11/2023	SoftEther proxy

193.36.119[.]45	IPv4	4/19/2023	7/7/2023	SoftEther proxy
185.158.248[.]159	IPv4	4/24/2023	7/6/2023	SoftEther proxy
131.153.78[.]188	IPv4	5/6/2023	6/29/2023	SoftEther proxy
37.143.130[.]146	IPv4	5/12/2023	5/19/2023	SoftEther proxy
146.70.157[.]45	IPv4	5/12/2023	6/8/2023	SoftEther proxy
185.195.200[.]39	IPv4	5/15/2023	6/29/2023	SoftEther proxy
185.38.142[.]229	IPv4	5/15/2023	7/12/2023	SoftEther proxy
146.70.121[.]44	IPv4	5/17/2023	6/29/2023	SoftEther proxy
31.42.177[.]181	IPv4	5/22/2023	5/23/2023	SoftEther proxy

185.51.134[.]52	IPv4	6/7/2023	7/11/2023	SoftEther proxy
173.44.226[.]70	IPv4	6/9/2023	7/11/2023	SoftEther proxy
45.14.227[.]233	IPv4	6/12/2023	6/26/2023	SoftEther proxy
185.236.231[.]109	IPv4	6/12/2023	7/3/2023	SoftEther proxy
178.73.220[.]149	IPv4	6/16/2023	7/12/2023	SoftEther proxy
45.14.227[.]212	IPv4	6/19/2023	6/29/2023	SoftEther proxy
91.222.173[.]225	IPv4	6/20/2023	7/1/2023	SoftEther proxy
146.70.35[.]168	IPv4	6/22/2023	6/29/2023	SoftEther proxy
146.70.157[.]213	IPv4	6/26/2023	6/30/2023	SoftEther proxy

31.42.177[.]201	IPv4	6/27/2023	6/29/2023	SoftEther proxy
5.252.176[.]8	IPv4	7/1/2023	7/1/2023	SoftEther proxy
80.85.158[.]215	IPv4	7/1/2023	7/9/2023	SoftEther proxy
193.149.129[.]88	IPv4	7/2/2023	7/12/2023	SoftEther proxy
5.252.178[.]68	IPv4	7/3/2023	7/11/2023	SoftEther proxy
116.202.251[.]8	IPv4	7/4/2023	7/7/2023	SoftEther proxy

Further reading

For the latest security research from the Microsoft Threat Intelligence community, check out the Microsoft Threat Intelligence Blog: https://aka.ms/threatintelblog.

To get notified about new publications and to join discussions on social media, follow us on Twitter at https://twitter.com/MsftSecIntel.