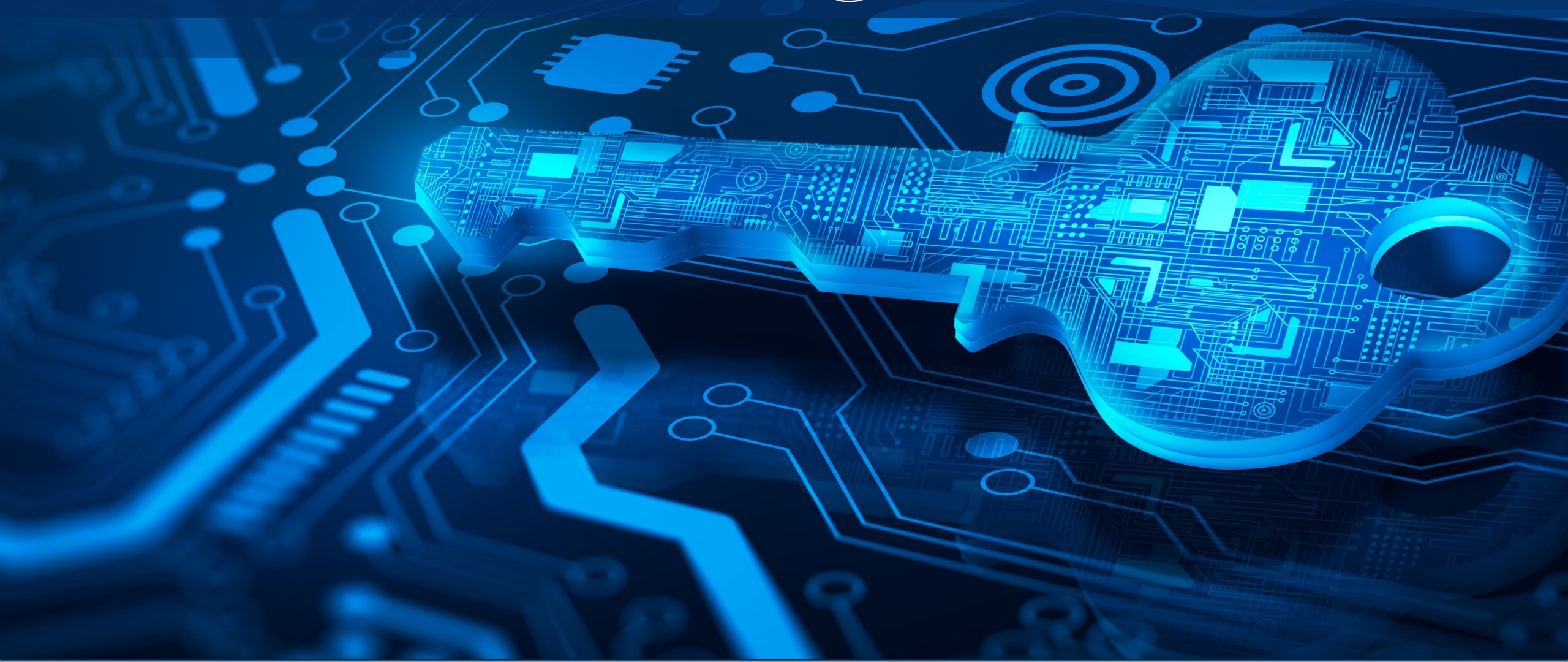


Private Keys in Public Places

Tom Pohl
Pentest Team Manager



whoami

- **Tom Pohl @tompohl**
- Pentest Team Manager, LMG Security
- Won CTF competitions at:
 - DEFCON
 - BSidesLV
 - THOTCON
 - 2 x Wild West Hackin' Fest
 - 3 x Circle City Con
- **An Evil Twin (Hi, Andy!)**



Hard at work!



Special Thanks!

- **Nick Starke** – My partner in crime (thought crimes) and why I even started looking for private keys
- **Sherri Davidoff, my wife** – this talk wouldn't have existed if it wasn't for her support and urging me to create it.



Roadmap

- How this started
- Netgear
- Fortinet
- Dell Compellent
- Recommendations



Why Should We Care About Private Keys?

- Private key exposure isn't the new sexy vulnerability
- It's also a hard software development problem
- In the real world..
 - Hackers take the easy path
 - Often way easier than a CTF
- We need to raise awareness!



“So in war, the way is to avoid what is strong, and strike at what is weak” – Sun Tzu, *The Art of War*



Once Upon a Time...

- In January 2020...
- My buddy Nick Starke reached out
- He was angry at Netgear
- Reported a bunch of bugs through bug bounty program
- They were trying to combine all the bugs into an arbitrary "systemic issue"
 - Basically, screw him out of money
- I said: "Let's go find something to hack on, then"
- He said, "What should we hack on?"
- (the answer is obvious)



Hint: Netgear

- Let's go pull down firmware
- See what we find!
- After just a few firmware downloads...

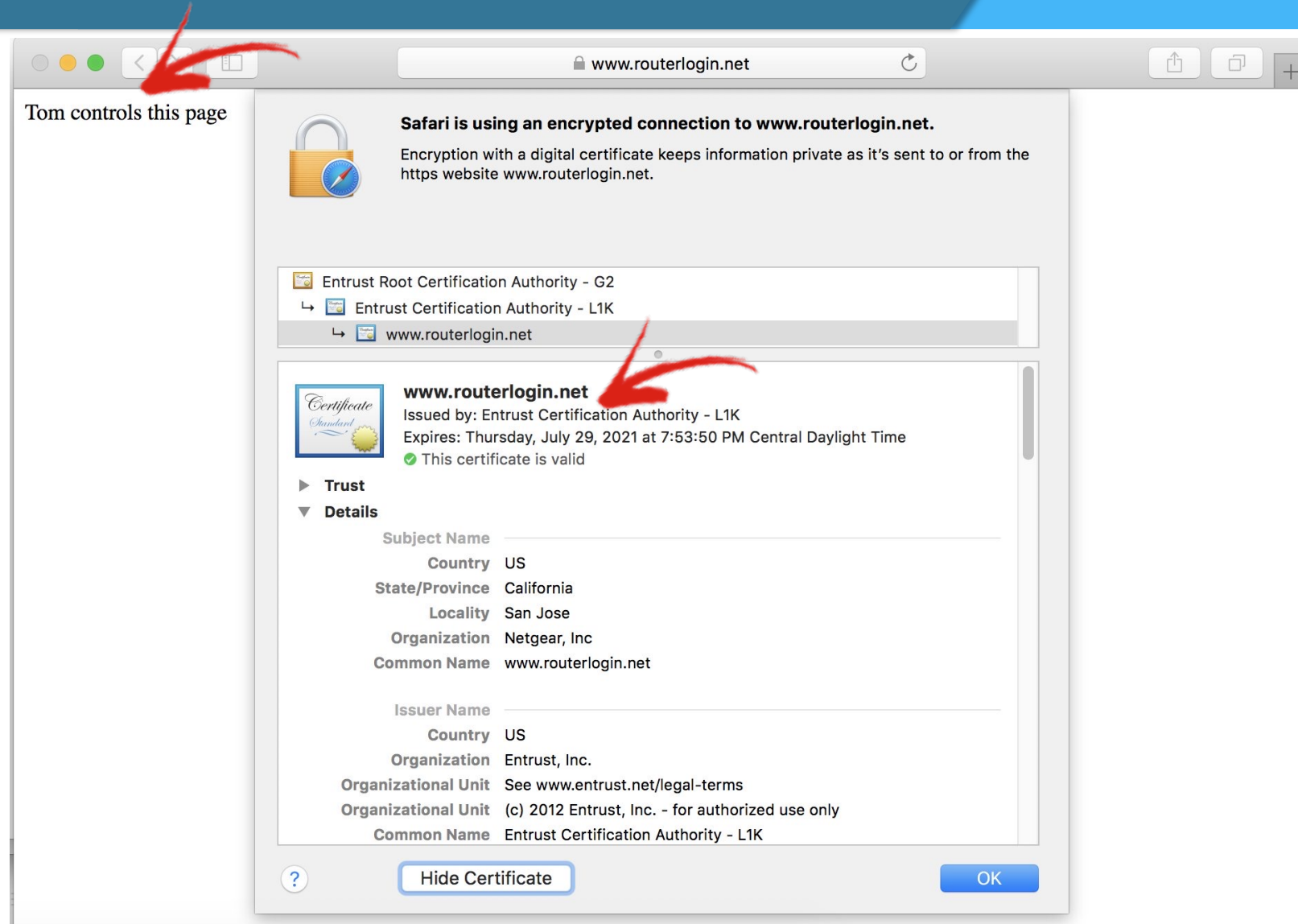
We found TWO different CA-blessed public certificates with the corresponding PRIVATE keys

routerlogin.net (Entrust)
and
mini-app.funjsq.com (COMODO)



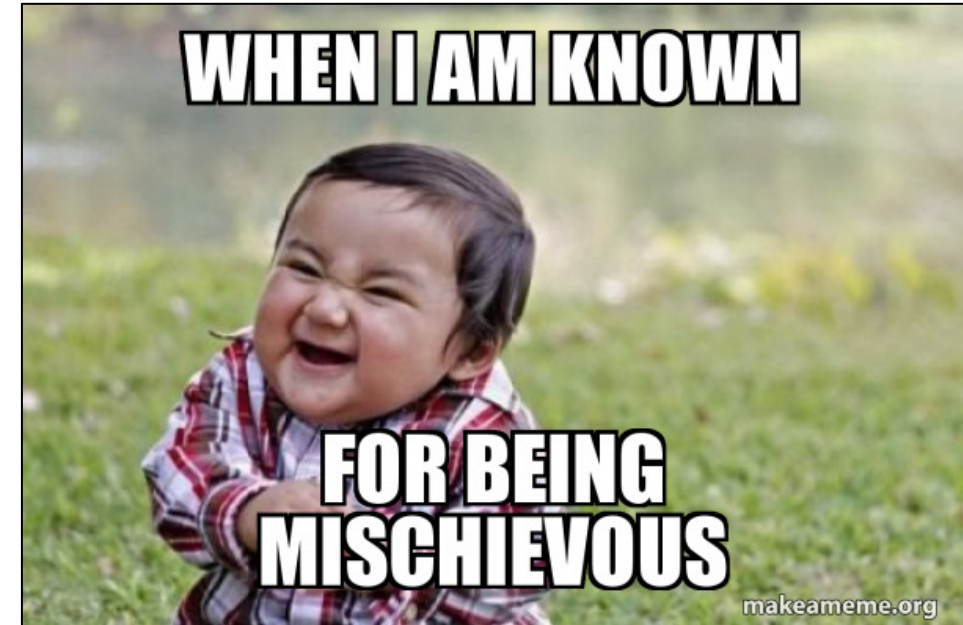
Proof that We Have the Private Keys

- Here we're using a CA-signed private key
- Created our own trusted web page
- routerlogin.net (Entrust)
- (Note: I control the DNS)



If I Were An Attacker (Which I Am, Professionally)...

- I would sit in a coffeeshop
 - Set up a fake DNS server
 - Force people to go to routerlogin.net
 - (since I have the private key to this site)
 - Looks legit!
 - If they have saved creds from logging into their router, they will give them to me!
 - This works anywhere I can poison DNS
- If remote administration is enabled, log in remotely and manage their network for them
- Target users for credential stuffing (due to password re-use)



We Tried to Tell Netgear

- Jan 14 2020 - Initial discovery
- Jan 14, 2020 – Sent a DM to Netgear on Twitter to establish communications
- Jan 15, 2020 – Reached out to Bugcrowd to attempt to establish communications
- Jan 16, 2020 – Bugcrowd responds and says we must submit through their bug bounty program. Refuses to accept information about the issue outside of this.
 - We're not interested. Here's why...



See Nick's writeup: <https://gist.github.com/nstarke/a611a19aab433555e91c656fe1f030a9>



The Insidious “Responsible Disclosure”

- They don't have to fix it
 - They'll pay you hush money
 - And then you can't say anything
- We didn't take Netgear's money
 - “Coordinated disclosure”
- Netgear wouldn't talk to us without going through the bug bounty program (which would force us to sign away our right to disclose)
- So we gave up and publicly disclosed
 - Then they came up with a fix within a few days!



“Bug Bounty Programs are Being Used to Buy Silence” – Bruce Schneier, April 3, 2020
https://www.schneier.com/blog/archives/2020/04/bug_bounty_prog.html



What Did We Learn from Netgear?

- Private keys can be trivially found
- We started to wonder...
- Where else might we find private keys?



What Else Might Have Private Keys?

- Let's look!
- I was a Fortinet customer at the time
- Decided to download the latest version of our FortiOS that our firewalls used
- Just take a little look to see what's in there



Inside the Firmware...

```
[root@tpohl-squid:~/fortinet2020/rootfs_dir# ls -l
total 30788
-rw-r--r-- 1 root root 24657844 Jul 19 07:47 bin.tar.xz
drwxr-xr-x 2 root root 4096 Jul 19 07:47 boot
drwxr-xr-x 2 root root 4096 Jul 19 07:47 data
drwxr-xr-x 2 root root 4096 Jul 19 07:47 data2
drwxr-xr-x 6 root root 24576 Jul 19 07:47 dev
lrwxrwxrwx 1 root root 8 Jul 19 07:47 etc -> data/etc
lrwxrwxrwx 1 root root 1 Jul 19 07:47 fortidev -> /
lrwxrwxrwx 1 root root 10 Jul 19 07:47 init -> /sbin/init
drwxr-xr-x 3 root root 4096 Jul 19 07:47 lib
-rw-r--r-- 1 root root 5828456 Jul 19 07:47 migadmin.tar.xz
drwxr-xr-x 2 root root 4096 Jul 19 07:47 proc
drwxr-xr-x 2 root root 4096 Jul 19 07:47 sbin
drwxr-xr-x 5 root root 4096 Jul 19 07:47 security-rating
drwxr-xr-x 2 root root 4096 Jul 19 07:47 sys
drwxr-xr-x 2 root root 4096 Jul 19 07:47 tmp
drwxr-xr-x 3 root root 4096 Jul 19 07:47 usr
-rw-r--r-- 1 root root 966740 Jul 19 07:47 usr.tar.xz
drwxr-xr-x 8 root root 4096 Jul 19 07:47 var
[root@tpohl-squid:~/fortinet2020/rootfs_dir# file *.xz
```

```
bin.tar.xz:      XZ compressed data
migadmin.tar.xz: XZ compressed data
usr.tar.xz:      XZ compressed data
```

XZ compressed data you say ...



What Happens When You Try to Decompress?

```
# xz -d bin.tar.xz
```

```
xz: bin.tar.xz: Compressed data is corrupt
```



Strange Things are Afoot at the Circle K^W^W^W Fortinet

- But it's not really corrupt...
- It's a brand new firmware downloaded directly from Fortinet
- I intentionally downloaded the x86 64-bit version of their firmware
- They had binaries on the firmware
- Could run them when I created my own chroot environment



How to Extract the “Corrupt” files ...

Since I downloaded the x86-64 VM, I can use the binaries in the firmware to extract / decompress the files

```
# cd ~/fortinet2020/rootfs_dir
# ls -l sbin
total 176
-rwxr-xr-x 1 root root 22936 Jul 19 07:47 ftar
-rwxr-xr-x 1 root root 10888 Jul 19 07:47 init
-rwxr-xr-x 1 root root 141832 Jul 19 07:47 xz
# ldd sbin/ftar
linux-vdso.so.1 (0x00007fff719a4000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f9e02845000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f9e02653000)
/fortidev/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 => /lib64/ld-linux-x86-64.so.2 (0x00007f9e029b5000)
# ldd sbin/xz
linux-vdso.so.1 (0x00007ffd53392000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fd61ac8f000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd61aa9d000)
/fortidev/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 => /lib64/ld-linux-x86-64.so.2 (0x00007fd61adff000)
# sbin/xz -d bin.tar.xz
-bash: sbin/xz: No such file or directory
```



```
# chroot /root/fortinet2020/rootfs_dir/ sbin/xz -d bin.tar.xz
# chroot /root/fortinet2020/rootfs_dir/ sbin/ftar -xf bin.tar
```



Let's Go Exploring!

- Now I go look for private keys again
- Same technique as before

- TWO different CA-blessed public certificates with the corresponding PRIVATE keys
- But this time...
 - Apple (APNS)
 - Google (GCM)



Here's the Apple Certificate

```
# openssl x509 -in apn-dis-cert.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 197504354987856146 (0x2bdad2a18a4d112)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, O = Apple Inc., OU = Apple Worldwide Developer Relations, CN = Apple
Worldwide Developer Relations Certification Authority
        Validity
            Not Before: Aug 29 22:38:56 2019 GMT
            Not After : Sep 27 22:38:56 2020 GMT
        Subject: UID = FortiToken-Mobile, CN = Apple Push Services: FortiToken-Mobile, OU =
AH4XFXJ7DK, O = "Fortinet, Inc", C = US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public-Key: (2048 bit)
            Modulus:
                00:a4:b0:aa:49:8b:da:cb:1b:65:4b:2f:2b:bf:9d:
                e4:43:87:07:28:b3:c8:68:84:ce:85:46:14:7f:07:
                b2:e7:7b:f5:9b:3f:d9:c1:1b:14:65:02:bf:3b:ca:
                5c:28:20:ba:19:e7:38:ed:58:38:2f:e5:58:3e:17:
                0c:c5:63:96:53:27:1d:5c:17:80:8a:23:43:a1:e8:
                b4:6f:aa:d5:bf:df:2b:0d:5f:72:08:81:34:ef:31:
```



I Have the Private Key!

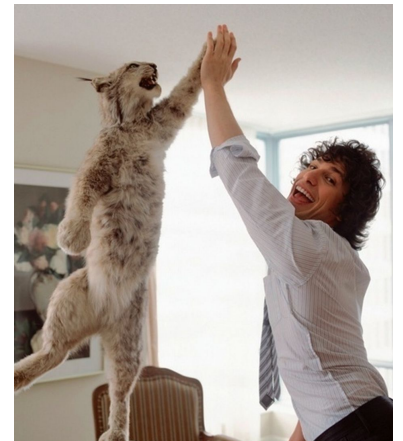
- The Apple certificate:

```
# openssl x509 -noout -modulus -in apn-dis-cert.pem | openssl md5  
(stdin)= 059b3ab6983af4509a8e09886f1a03e1
```

- Corresponding Private Key:

```
# openssl rsa -noout -modulus -in apn-dis-key.pem | openssl md5  
(stdin)= 059b3ab6983af4509a8e09886f1a03e1
```

It's a Match!



If I Were An Attacker...

- According to Fortinet, these keys are “used to establish secure communication with both Apple Push Notification and Google Cloud Messaging services”
- Does this mean I could send push notifications to Fortinet users as Fortinet?
- (At this point, I decided to report to Fortinet)



Reporting to Fortinet

- 1/28/2020: Reported to their Product Security Incident Response Team via email
- 1/28/2020: Emailed Apple to report
- 1/29/2020: Apple quick response asking me to provide the certificate for revocation
- 1/29/2020: Fortinet responded after some hiccups with their GPG encrypted email
- 1/31/2020: As requested by Apple, I sent the certificate
- 2/5/2020: Fortinet said it would be fixed in an upcoming release.
 - Assigned CVE-2020-6645
- 4/20/2020: Fix probably coming in FortiOS 6.4.1 and asked me to wait to disclose
- 4/20/2020: I agreed!



The Big Wait



Three Years Later...



NEWS / RESEARCH SERVICES THREAT INTELLIGENCE RESOURCES

► Home / PSIRT / FG-IR-20-014



PSIRT Advisories

FortiOS, FortiAuthenticator - Disclosure of private keys corresponding to Apple (APNS) and Google (GCM) certificates

Summary

A clear text storage of sensitive information (CWE-312) vulnerability in both FortiGate and FortiAuthenticator may allow a local unauthorized party to retrieve the Fortinet private keys used to establish secure communication with both Apple Push Notification and Google Cloud Messaging via accessing the files on the filesystem.

The potentially exposed private keys have been revoked, please upgrade to the versions provided in the table below to support push proxy.

Affected Products

FortiOS version 6.4.0 through 6.4.1
FortiOS version 6.2.0 through 6.2.9
FortiOS version 6.0.0 through 6.0.13
FortiAuthenticator version 6.1.0
FortiAuthenticator version 6.0.0 through 6.0.4
FortiAuthenticator 5.5 all versions

Date	Feb 16, 2023
Severity	● ● ● ● ● Medium
CVSSv3 Score	5.3
Impact	Information disclosure
CVE ID	CVE-2022-22302
Affected Products	FortiAuthenticator : 6.1.0, 6.0.4, 6.0.3, 6.0.2, 6.0.1, 6.0.0, 5.5.0 FortiOS : 6.4.1, 6.4.0, 6.2.9, 6.2.8, 6.2.7, 6.2.6, 6.2.5, 6.2.4, 6.2.3, 6.2.2, 6.2.1, 6.2.0, 6.0.9, 6.0.8, 6.0.7, 6.0.6, 6.0.5, 6.0.4, 6.0.3, 6.0.2, 6.0.13, 6.0.12, 6.0.11, 6.0.10, 6.0.1, 6.0.0

<https://www.fortiguards.com/psirt/FG-IR-20-014>

But Wait... It's Not Over Yet

- We were on an external penetration test
- Target had a Fortinet firewall w/ VPN
- Password-guessing attacks worked!
 - But MFA was turned on
- Could we bypass the MFA somehow?
- Decided to download the firmware to inspect
- ...And that was when I found the FIX to the original issue



The Forti-Fix

- Spelunking through the massive init file in Fortinet firmware...
- Discovered a URL for doing push notifications - productapi.fortinet.com
- The URL required a client certificate

- Found a certificate with the corresponding PRIVATE key
- But this time...
 - Fortinet-issued certificate
- We validated that this was a client certificate to talk to the Fortinet endpoint that sends push notification

```
fortiOS
notificationtype
%s()-%d: CURL init error.
Accept: application/ison
Charsets: utf-8
/etc/fgt.crt
/etc/fgt.key
%s()-%d: iOS push data:
productapi.fortinet.com
URL: %s
Message: %s

action plugin stats:
flags:%x

triggertype
rawlog
devname
Host: %s
%s:443:%s
%s()-%d: CSF error:%s
%s()-%d: server path:%s
unknown:%d
```



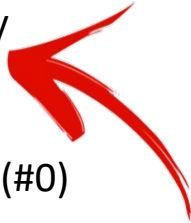
We Oh for Forti-Sake

```
[root@tpohl-squid:~/fortinet/rootfs_uncpio/tmp/etc# curl https://productapi.fortinet.com -vk
* Trying 209.52.38.133:443...
* TCP_NODELAY set
* Connected to productapi.fortinet.com (209.52.38.133) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
* CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Request CERT (13):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS alert, handshake failure (552):
* error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake failure
* Closing connection 0
curl: (35) error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake failure
```



The New Found Key Works ...

```
# curl -vk https://productapi.fortinet.com --cert fgt.crt --key fgt.key
* Trying 209.52.38.133:443...
* TCP_NODELAY set
* Connected to productapi.fortinet.com (209.52.38.133) port 443 (#0)
...
* Server certificate:
* subject: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=MIS; CN=productapi.fortinet.com; emailAddress=support@fortinet.com
* start date: May 1 16:53:32 2023 GMT
* expire date: Apr 30 16:53:32 2024 GMT
* issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=support; emailAddress=support@fortinet.com
* SSL certificate verify result: self signed certificate in certificate chain (19), continuing anyway.
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* Using Stream ID: 1 (easy handle 0x55a049a71300)
> GET / HTTP/2
> Host: productapi.fortinet.com
> user-agent: curl/7.68.0
> accept: */*
```



- I didn't get a chance to play with it any more to see if there's other authentication methods in play (let's hope!)



What Did We Learn from Fortinet?

- Private keys are everywhere
- Manufacturers may not prioritize this type of issue
- It's an ARCHITECTURAL issue (not easy to fix)
- Can take a really long time to address



We Kept Finding Private Keys

- They are very useful on pentests!
- Fast forward to April 2023
- My team was conducting an internal penetration test
 - Our computer was plugged into the equivalent of a conference room
 - No corporate user ID or password
- Scanned the network
- Conducted a password-guessing attack
- Logged into an SSH server
 - Can you guess the password?

admin/admin



Exploring the Mystery Device

- Linux-based machine
- Running some Java
- Found some properties files that had user creds to the vCenter environment
 - They were encrypted
- How can we decrypt them?



Looking Through Running Processes

- Discovered where the software was that reads/writes configs

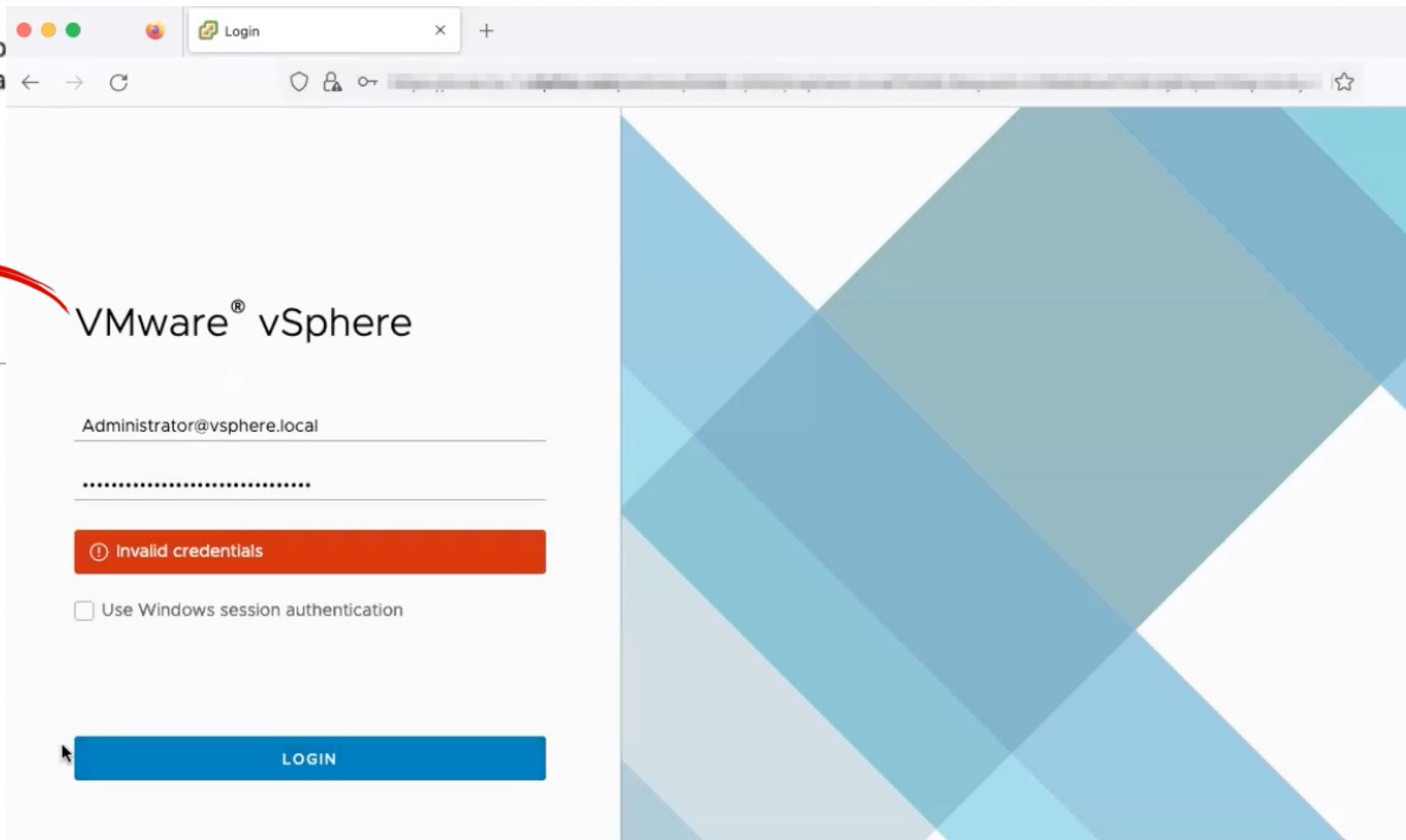
```
[admin@cvit ~]$ ps auxw
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2072    632 ?        Ss   2020    0:01 init [3]
root         2  0.0  0.0     0      0 ?        S<   2020    0:00 [migration/0]
root         3  0.0  0.0     0      0 ?        SN   2020    0:00 [ksoftirqd/0]
root         4  0.0  0.0     0      0 ?        S<   2020    0:00 [watchdog/0]
root         5  0.0  0.0     0      0 ?        S<   2020    0:01 [events/0]
root         6  0.0  0.0     0      0 ?        S<   2020    0:00 [khelper]
root         7  0.0  0.0     0      0 ?        S<   2020    0:00 [kthread]
root        10  0.0  0.0     0      0 ?        S<   2020    0:00 [kblockd/0]
root        11  0.0  0.0     0      0 ?        S<   2020    0:00 [kacpid]
root       161  0.0  0.0     0      0 ?        S<   2020    0:00 [cqueue/0]
root       164  0.0  0.0     0      0 ?        S<   2020    0:00 [khubd]
root       166  0.0  0.0     0      0 ?        S<   2020    0:00 [kseriod]
root       232  0.0  0.0     0      0 ?        S   2020    0:00 [khungtaskd]
root       233  0.0  0.0     0      0 ?        S   2020    0:00 [pdflush]
root       234  0.0  0.0     0      0 ?        S   2020    0:01 [pdflush]
root       235  0.0  0.0     0      0 ?        S<   2020    0:00 [kswapd0]
root       236  0.0  0.0     0      0 ?        S<   2020    0:00 [aio/0]
root       449  0.0  0.0     0      0 ?        S<   2020    0:00 [kpsmoused]
root       479  0.0  0.0     0      0 ?        S<   2020    0:00 [mpt_poll_0]
root       480  0.0  0.0     0      0 ?        S<   2020    0:00 [mpt/0]
root       481  0.0  0.0     0      0 ?        S<   2020    0:00 [scsi_eh_0]
root       484  0.0  0.0     0      0 ?        S<   2020    0:00 [ata/0]
root       485  0.0  0.0     0      0 ?        S<   2020    0:00 [ata_aux]
root       492  0.0  0.0     0      0 ?        S<   2020    0:00 [ata_aux]
root       501  0.0  0.0     0      0 ?        S<   2020    4:3 [kauditd]
root       526  0.0  0.0     0      0 ?        S<   2020    0:00 [kauditd]
tomcat    2921  0.0  2.2 456840 48044 ?        Sl   2020 140:28 /usr/java/jre1.6.0_27/bin/java -classpath :/usr/share/tomcat6/bin/bootstrap.jar:/usr/share
llent/rmsv/axis2: opt/dellcompellent/rmsv/lib /opt/dellcompellent/rmsv/endorsed/ -Dcatalina.base=/opt/dellcompellent/rmsv -Dcatalina.home=/usr/share/tomca
/dellcompellent/rmsv/temp -Djava.util.logging.config.file=/opt/dellcompellent/rmsv/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.Cla
tomcat    2992  0.0  4.6 435400 97308 ?        Sl   2020 61:44 /usr/java/jre1.6.0_27/bin/java -classpath :/usr/share/tomcat6/bin/bootstrap.jar:/usr/shar
e=/usr/share/tomcat6 -Dcatalina.home=/usr/share/tomcat6 -Djava.endorsed.dirs= -Djava.io.tmpdir=/var/cache/tomcat6/temp -Djava.util.logging.config.file=/us
juli.ClassLoaderLogManager org.apache.catalina.startup.Bootstrap start
```

The directory of the Dell Compellent software



Found a Properties File With Credentials(?!)

```
[root@cvit ~]# cd /opt/dellcompellent/  
[root@cvit dellcompellent]# ls  
lib  rmsv  vsp  
[root@cvit dellcompellent]# find . |grep vsp.properties  
./vsp/vsp.properties  
[root@cvit dellcompellent]# cat ./vsp/vsp.prop  
#Dell Compellent VMware vSphere Plugin for VMwa  
#Wed Oct 15 15:45:35 PDT 2014  
HostName=  
Password= 5DE752DB122F2DB72F310D2A3992666E  
PackageName=vcenterplugin2.0.1-1.zip  
Version=2.0.1.1  
UserName=vsphere.local\\administrator  
WebHostName=  
[root@cvit dellcompellent]#
```



VMware® vSphere

Administrator@vsphere.local

.....

Invalid credentials

☐ Use Windows session authentication

LOGIN



Let's Find The Key!

Step 1 - Find the software that can read/write the config

```
echo "Configuring Dell Compellent vSphere Web Client Plugin..."  
/usr/java/jre1.6.0_27/bin/java -DRMSV_HOME=$RMSV_HOME -DVSP_HOME=/opt/dellcompellent/vsp -classpath /opt/dellcompellent/vsp/VSPConfigUtil.jar  
:/opt/dellcompellent/vsp/lib/vijava-osi.jar com.dell.compellent.vsp.config.util.GetConfigurationData -webhostname $MGMT_IPADDR  
# sometimes Tomcat is dead at this point. Make sure it is running.  
startRMTomcat
```

`-classpath /opt/dellcompellent/vsp/VSPConfigUtil.jar`

`com.dell.compellent.vsp.config.util.GetConfigurationData`



Let's Find the Key!

- Step 2 - Unzip the Jar file

```
# unzip VSPConfigUtil.jar
Archive:  VSPConfigUtil.jar
  creating: META-INF/
  inflating: META-INF/MANIFEST.MF
  creating: com/
  creating: com/dell/
  creating: com/dell/compellent/
  creating: com/dell/compellent/vsp/
  creating: com/dell/compellent/vsp/config/
  creating: com/dell/compellent/vsp/config/util/
  inflating: com/dell/compellent/vsp/config/util/ConfUtil.class
  inflating: com/dell/compellent/vsp/config/util/GetConfigurationData$1.class
  inflating: com/dell/compellent/vsp/config/util/GetConfigurationData$AnyHostnameVerifier.class
  inflating: com/dell/compellent/vsp/config/util/GetConfigurationData$SavingTrustManager.class
  inflating: com/dell/compellent/vsp/config/util/GetConfigurationData.class
  inflating: com/dell/compellent/vsp/config/util/RegisterTaskHandler.class
  inflating: com/dell/compellent/vsp/config/util/VCRegister.class
  inflating: com/dell/compellent/vsp/config/util/WebThumbprint.class
```



Let's Find the Key!

- Step 3: Download a decompiler

```
# wget https://www.benf.org/other/cfr/cfr-0.152.jar
```

- Step 4: Decompile classes until you find the key! (I found it on my first try)

```
# java -jar cfr-0.152.jar ConfUtil.class
```



Decompiled the Software and Found...

```
public class ConfUtil {  
    public static final String VC_HOST_NAME = "HostName";  
    public static final String VC_USER_NAME = "UserName";  
    public static final String VC_PASSWORD = "Password";  
    public static final String VC_WEBHOST_NAME = "WebHostName";  
    public static final String VC_VERSION = "Version";  
    public static final String VC_PACKAGENAME = "PackageName";  
    private static String[] packageKeys = new String[]{"PackageName", "Version"};  
    private static String[] credentialKeys = new String[]{"HostName", "UserName", "Password", "WebHostName"};  
    private static final String ENCRYPTION_TYPE = "AES";  
    private static final String ENCRYPTION_KEY = "Recovered AES Key";  
    public static final String FILE_SEPARATOR = System.getProperty("file.separator");  
    public static final String VSP_HOME;  
    public static final String CONF_DIR_NAME;  
    private static final String CONF_FILE_NAME = "vsp.properties";  
    private static final String PACKAGE_FILE_NAME = "pkg.properties";  
    private static final String PROPERTIES_FILE_NAME;  
    private static final String PACKAGEINFO_FILE_NAME;
```

Recovered AES Key



- A static AES key
- Used it to decrypt the admin credentials to the vCenter server



With the AES Key – we recovered the vCenter creds

```
[root@cvit ~]# cd /opt/dellcompellent/  
[root@cvit dellcompellent]# ls  
lib  rmsv  vsp  
[root@cvit dellcompellent]# find . |grep vsp.properties  
./vsp/vsp.properties  
[root@cvit dellcompellent]# cat ./vsp/vsp.properties  
#Dell Compellent VMware vSphere Plugin for VMware vCenter configuration parameters...  
#Wed Oct 15 15:45:35 PDT 2014  
HostName=██████████  
Password=5DE752DB122F2DB72F310D2A3992666E  
PackageName=vcenterplugin2.0.1-1.zip  
Version=2.0.1.1  
Username=vsphere.local\\administrator  
WebHostName=██████████  
[root@cvit dellcompellent]#
```

Download CyberChef [↓](#) Last build: 16 days ago - Version 10 is here! Read about the new feat... Options [⚙](#) About / Support

Operations	Recipe	Input
aes ×	AES Decrypt ⏏ ⏏	5DE752DB122F2DB72F310D2A3992666E
AES Decrypt	Key HEX	
AES Encrypt	IV HEX	
AES Key Wrap	Mode	
AES Key Unwrap	Input	
Parse ASN.1 hex string	CBC	Hex
Group IP addresses	Output	
Parse IPv6 address	Raw	
Defang IP Addresses		
Generate all hashes		
Extract IP addresses		

STEP **BAKE!** ☒ Auto Bake

rec 32 1 Raw Bytes LF

Output

NotThePassword

rec 14 1 9ms Raw Bytes LF



Attackers Don't Break in, They Login

New vCenter server updates are available [VIEW UPDATES](#)

vSphere Client

Summary Monitor Configure Permissions Datacenters Hosts & Clusters VMs Datastores Networks

Version: 7.0.3 **Updates Available**
Build: 18700403
Last Updated: Nov 3, 2022, 11:49 AM
Last File-Based Backup: Not scheduled

Clusters: 5
Hosts: 12
Virtual Machines: 141

CPU Free: 858.16 GHz
Used: 100.62 GHz Capacity: 958.78 GHz
Memory Free: 5.63 TB
Used: 2.12 TB Capacity: 7.75 TB
Storage Free: 65.02 TB
Used: 67.02 TB Capacity: 132.03 TB

Virtual machine CPU usage [Acknowledge](#) [Reset To Green](#)

Custom Attributes

Attribute	Value
-----------	-------

Health Status

Overall Health Good

Recent Tasks Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time	Completion
-----------	--------	--------	---------	-----------	------------	------------	------------

All [More Tasks](#) 0 items



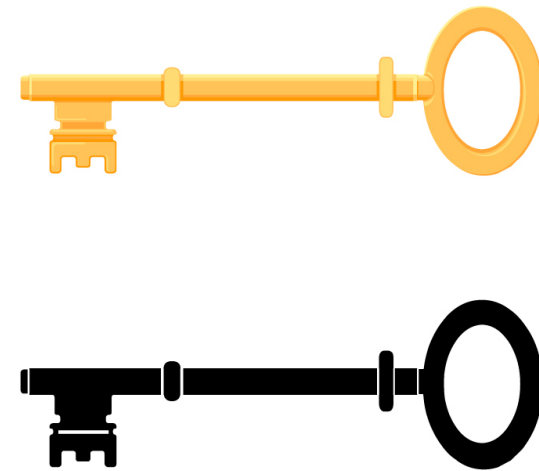
Game Over!

- Normally, we'd snapshot a domain controller and forensically recover the NTDS.dit...
 - But we didn't have to!
- While my team was looking through vCenter consoles...
- I found the Certificate Authority (testing with certipy)
 - Network admin got an alert
- One of their admins logged in to see what was going on
 - And then walked away
 - Left the console unlocked
- My team saw and quickly added a domain admin account



The Same Key Is Used for Every Customer!

- Two ways to exploit this:
 1. Default credentials – password-guessing attack
 2. User with any level of access to Dell Storage Compellent Integration Tools for VMware can escalate privileges and take over vCenter
- Think: insider threat or low-level compromise



Reported to Dell

- 4/11/2023 – Emailed Dell w/ 90-day disclosure deadline
- 4/14/2023 – Explained that I'm NOT a customer and I can't open a support ticket
- 5/11/2023 – Dell responded
 - They said "not a problem" – default creds
- 5/13/2023 – I responded explaining the situation a little more
 - Talked about segregation of duties and all levels of access could perform the attack
- 6/29/2023 – Got an update
 - They're fixing it!
 - In November



There are LOTS More Private Keys Out There

- WAY more than this
 - Remote login apps
 - Routers
 - Cameras
 - Printers (see my talk “How I Met Your Printer”)
 - <https://www.youtube.com/watch?v=xaCilyQ5QXM>
 - & more
- **This is a systemic problem!**



Recommendations

1. Firmware and software developers need to remove private keys and encryption keys from their firmware and software binaries
2. Regular security audits and testing should be done to check for hidden secrets
3. Coordinated disclosure programs should be adopted by companies to allow for responsible disclosure of vulnerabilities without the need for public releases



Questions?

- Tom Pohl, Pentest Team Manager
- info@LMGsecurity.com



@tompohl



tompohl@infosec.exchange

- Find me on **LinkedIn**





Please contact us any time you have a question or need additional support.

1-855-LMG-8855

info@LMGsecurity.com