TLP: GREEN
11.12.2015

Case Study: FastDNS Infrastructure battles Xor Botnet

1.0 / OVERVIEW / Xor, a Trojan malware attackers are using to hijack Linux machines to include within a botnet for distributed denial of service (DDoS) campaigns, appears to be behind an Oct. 13 DNS flood against a customer using Akamai's FastDNS infrastructure.

This attack campaign started with a DNS Flood of 30 million queries per second and escalated into a Tsunami SYN Flood ramping up to 140 Gbps with over 75 Mpps in total. All attack signatures match with the recently investigated Xor.DDoS Botnet.

Between Oct. 13 and 23rd, the attack was constantly switching on and off. The attack hit multiple destination hosts at the same time.

Below is an outline of the attack characteristics and defensive measures as observed by Akamai's Security Intelligence Research Team (SIRT).

In the course of this investigation, the SIRT worked with Akamai's FastDNS team. The FastDNS team noticed considerable attack traffic while XoR was not targeted at us. It's possible the adversary is employing a multi-vendor DDoS approach and that all the DNS traffic we see is attributable to XoR. That said, we are reasonably certain that Xor is behind all the SYN flood activity.

2.0 / ATTACK CHARACTERISTICS / As explained in Section 3.0 of the Xor Threat Advisory released in late September, SYN and DNS floods generated by the Xor.DDoS Malware have very specific characteristics.

First, a SYN flood always consists of the following 8 static TCP header options.

```
0204 05b4 0101 0402
```

Figure 1: Static TCP headers

```
15:55:58.842287 IP xxx.187.0.215.51188 > y.y.y.y.53: Flags [S], seq 3354679860:3354680867, win
65535, length 10071460+ [b2&3=0x101] [1026q] Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type20828
(Class 0)? . Type0 (Class 36864)? M-u^I^IM-XC#^I^@^@^@^@^@^@^@^@M-8^ZM- @5^@^M^H ^@^@^@
^@^@^@^@^@^@^@^A^@^@^@^@M-c#^I^W^D^@^@M-8^ZM-!A5^@^@^@ M-P#^IM-^A^@^@^@(^Q$^IHM-O#^I M-
g#^I^W^D^@^@M-Al[M-^Z5^@^@5@M-P#^Ia^@^@^@Hj"^IM-^XM-V#^I@M-k#^I^W^D^@^@^W=M-GC5^@^@^@ ^@^@^@
^@^@^@^@^@^@^@^A^@^@^@`M-o#^I^W^D^@^@_dM-.B5^@^@^@M-^@M-P#^I!^@^@^@M-(Q4^Ih($^IM-^@M-
s#^I^W^D^@^@`^G1C5^@^@^@M-@. MX (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
20507)? . Type256 (Class 0)? [|domain]
        0x0000:  4520 0417 c7f4 0000 8606 30df xxbb 00d7  E.........0.....
        0x0010:  b81a a040 c7f4 0035 c7f4 5a34 0000 0000  ...@...5..Z4....
        0x0020:  5002 ffff ada5 0000 0204 05b4 0101 0402  P...............
        0x0030:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0040:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0050:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0060:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0070:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0080:  0000 0000 0000 0000 0000 0000 515c 0000  ............Q\..
        0x0090:  b8f5 0909 d843 2309 0000 0000 0000 0000  .....C#.........
        0x00a0:  b81a a040 3500 0d08 2000 0000 2000 0000  ...@5...........
        0x00b0:  0000 0000 0100 0000 00e3 2309 1704 0000  ..........#.....
        0x00c0:  b81a a141 3500 0000 20d0 2309 8100 0000  ...A5.....#.....
```

```
        0x00d0:  2811 2409 48cf 2309 20e7 2309 1704 0000  (.$.H.#...#.....
        0x00e0:  c16c 5b9a 3500 0035 40d0 2309 6100 0000  .l[.5..5@.#.a...
        0x00f0:  486a 2209 98d6 2309 40eb 2309 1704 0000  Hj"...#.@.#.....
        0x0100:  173d c743 3500 0000 2000 0000 2000 0000  .=.C5...........
        0x0110:  0000 0000 0100 0000 60ef 2309 1704 0000  ........`.#.....
        0x0120:  5f64 ae42 3500 0000 80d0 2309 2100 0000  _d.B5.....#.!...
        0x0130:  a851 3409 6828 2409 80f3 2309 1704 0000  .Q4.h($...#.....
        0x0140:  6007 3143 3500 0000 c000 0000 9000 0000  `.1C5...........
        0x0150:  0f00 0000 0000 0000 0000 0000 0000 0000  ................
        0x0160:  501b f2f4 0100 0000 0000 0000 0000 0000  P...............
        0x0170:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0180:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0190:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01d0:  0000 0000 0000 0000 0000 0000 8100 0000  ................
        0x01e0:  f04a 0d08 6828 2409 a0f7 2309 1704 0000  .J..h($...#.....
        0x01f0:  b81a a040 3500 6809 2000 0000 2000 0000  ...@5.h.........
        0x0200:  a8d1 2309 0100 0000 c0fb 2309 1704 0000  ..#.......#.....
        0x0210:  b81a a141 3500 0000 4000 0000 2000 0000  ...A5...@.......
        0x0220:  c8d1 2309 0100 0000 e0ff 2309 1704 0000  ..#.......#.....
        0x0230:  c16c 5b9a 3500 0000 6000 0000 2000 0000  .l[.5...`.......
        0x0240:  1808 2409 0100 0000 0004 2409 1704 0000  ..$.......$.....
        0x0250:  173d c743 3500 0000 d001 0000 9000 0000  .=.C5...........
        0x0260:  0f00 0000 0000 0000 8019 0000 9000 0000  ................
        0x0270:  0f00 0000 0000 0000 0000 0000 0000 0000  ................
        0x0280:  50fb 5fed 0100 0000 0000 0000 0000 0000  P._.............
        0x0290:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02d0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02f0:  0000 0000 0000 0000 0000 0000 8119 0000  ................
        0x0300:  f8f2 2209 b820 2409 0000 0000 0000 0000  .."...$.........
        0x0310:  b81a a040 3500 2309 0000 0000 6119 0000  ...@5.#.....a...
        0x0320:  78f6 1909 2877 2309 0000 0000 0000 0000  x...(w#.........
        0x0330:  b81a a040 fdf0 0035 fdf0 7644 0000 0000  ...@...5..vD....
        0x0340:  5802 ffff b075 0000 0204 05b4 0101 0402  X....u..........
        0x0350:  5802 ffff cc7e 0000 0204 05b4 0101 0402  X....~..........
        0x0360:  5802 ffff b87c 0000 0204 05b4 0101 0402  X....|..........
        0x0370:  5002 ffff b327 0000 0204 05b4 0101 0402  P....'..........
        0x0380:  b844 2309 f8b8 2309 0000 0000 4108 0000  .D#...#.....A...
        0x0390:  48e0 2209 c852 4c09 0000 0000 0000 0000  H."..RL.........
        0x03a0:  5f64 ae42 3238 0035 3238 6438 0000 0000  _d.B28.528d8....
        0x03b0:  58c2 ffff ef24 0000 0204 05b4 0101 0402  X....$..........
        0x03c0:  173d c743 3500 0000 00cf 2309 2104 0000  .=.C5.....#.!...
        0x03d0:  98d7 2309 2853 5409 0000 0000 0000 0000  ..#.(ST.........
        0x03e0:  173d c743 0420 0035 0420 5350 0000 0000  .=.C...5..SP....
        0x03f0:  5002 ffff e180 0000 0204 05b4 0101 0402  P...............
        0x0400:  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure 2: Captured sample #1

```
15:55:58.842419 IP xxx.187.0.184.30838 > y.y.y.y.53: Flags [S], seq 2021059849:2021060856, win
65535, length 10071460+ [b2&3=0x101] [1026q] Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type20828
(Class 0)? . Type0 (Class 36864)? M-u^I^IM-XC#^I^@^@^@^@^@^@^@M-8^ZM- @5^@^M^H ^@^@^@
^@^@^@^@^@^@^@^A^@^@^@^@M-c#^I^W^D^@^@M-8^ZM-!A5^@^@^@ M-P#^IM-^A^@^@^@(^Q$^IHM-O#^I M-
g#^I^W^D^@^@M-Al[M-^Z5^@^@5@M-P#^Ia^@^@^@Hj"^IM-^XM-V#^I@M-k#^I^W^D^@^@^W=M-GC5^@^@^@ ^@^@^@
^@^@^@^@^@^@^@^A^@^@^@`M-o#^I^W^D^@^@_dM-.B5^@^@^@M-^@M-P#^I!^@^@^@M-(Q4^Ih($^IM-^@M-
s#^I^W^D^@^@`^G1C5^@^@^@M-@. MX (Class 0)? . Type0 (Class 0)? . Type0 (Class 0)? . Type0 (Class
20507)? . Type256 (Class 0)? [|domain]
        0x0000:  4520 0417 7876 0000 5f06 a77c xxbb 00b8  E...xv.._..|....
        0x0010:  b81a a040 7876 0035 7876 ed09 0000 0000  ...@xv.5xv......
        0x0020:  5002 ffff b9eb 0000 0204 05b4 0101 0402  P...............
```

```
0x0030:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0040:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0050:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0060:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0070:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0080:  0000 0000 0000 0000 0000 0000 515c 0000  ............Q\..
0x0090:  b8f5 0909 d843 2309 0000 0000 0000 0000  .....C#.........
0x00a0:  b81a a040 3500 0d08 2000 0000 2000 0000  ...@5...........
0x00b0:  0000 0000 0100 0000 00e3 2309 1704 0000  ..........#.....
0x00c0:  b81a a141 3500 0000 20d0 2309 8100 0000  ...A5.....#.....
0x00d0:  2811 2409 48cf 2309 20e7 2309 1704 0000  (.$.H.#...#.....
0x00e0:  c16c 5b9a 3500 0035 40d0 2309 6100 0000  .l[.5..5@.#.a...
0x00f0:  486a 2209 98d6 2309 40eb 2309 1704 0000  Hj"...#.@.#.....
0x0100:  173d c743 3500 0000 2000 0000 2000 0000  .=.C5...........
0x0110:  0000 0000 0100 0000 60ef 2309 1704 0000  ........`.#.....
0x0120:  5f64 ae42 3500 0000 80d0 2309 2100 0000  _d.B5.....#.!...
0x0130:  a851 3409 6828 2409 80f3 2309 1704 0000  .Q4.h($...#.....
0x0140:  6007 3143 3500 0000 c000 0000 9000 0000  `.1C5...........
0x0150:  0f00 0000 0000 0000 0000 0000 0000 0000  ................
0x0160:  501b f2f4 0100 0000 0000 0000 0000 0000  P...............
0x0170:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0180:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0190:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x01a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x01b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x01c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x01d0:  0000 0000 0000 0000 0000 0000 8100 0000  ................
0x01e0:  f04a 0d08 6828 2409 a0f7 2309 1704 0000  .J..h($...#.....
0x01f0:  b81a a040 3500 6809 2000 0000 2000 0000  ...@5.h.........
0x0200:  a8d1 2309 0100 0000 c0fb 2309 1704 0000  ..#.......#.....
0x0210:  b81a a141 3500 0000 4000 0000 2000 0000  ...A5...@.......
0x0220:  c8d1 2309 0100 0000 e0ff 2309 1704 0000  ..#.......#.....
0x0230:  c16c 5b9a 3500 0000 6000 0000 2000 0000  .l[.5...`.......
0x0240:  1808 2409 0100 0000 0004 2409 1704 0000  ..$.......$.....
0x0250:  173d c743 3500 0000 d001 0000 9000 0000  .=.C5...........
0x0260:  0f00 0000 0000 0000 8019 0000 9000 0000  ................
0x0270:  0f00 0000 0000 0000 0000 0000 0000 0000  ................
0x0280:  50fb 5fed 0100 0000 0000 0000 0000 0000  P._.............
0x0290:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02d0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x02f0:  0000 0000 0000 0000 0000 0000 8119 0000  ................
0x0300:  f8f2 2209 b820 2409 0000 0000 0000 0000  .."...$.........
0x0310:  b81a a040 3500 2309 0000 0000 6119 0000  ...@5.#.....a...
0x0320:  78f6 1909 2877 2309 0000 0000 0000 0000  x...(w#.........
0x0330:  b81a a040 fdf0 0035 fdf0 7644 0000 0000  ...@...5..vD....
0x0340:  5802 ffff b075 0000 0204 05b4 0101 0402  X....u..........
0x0350:  5802 ffff cc7e 0000 0204 05b4 0101 0402  X....~..........
0x0360:  5802 ffff b87c 0000 0204 05b4 0101 0402  X....|..........
0x0370:  5002 ffff b327 0000 0204 05b4 0101 0402  P....'..........
0x0380:  b844 2309 f8b8 2309 0000 0000 4108 0000  .D#...#.....A...
0x0390:  48e0 2209 c852 4c09 0000 0000 0000 0000  H."..RL.........
0x03a0:  5f64 ae42 3238 0035 3238 6438 0000 0000  _d.B28.528d8....
0x03b0:  58c2 ffff ef24 0000 0204 05b4 0101 0402  X....$..........
0x03c0:  173d c743 3500 0000 00cf 2309 2104 0000  .=.C5.....#.!...
0x03d0:  98d7 2309 2853 5409 0000 0000 0000 0000  ..#.(ST.........
0x03e0:  173d c743 0420 0035 0420 5350 0000 0000  .=.C...5..SP....
0x03f0:  5002 ffff e180 0000 0204 05b4 0101 0402  P...............
0x0400:  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0410:  0000 0000 0000 00                        .......
```

Figure 3: Captured sample #2

We see a 100% duplicate payload, confirming the spoofing capabilities explained in section 4 of the Xor advisory.

The two packets are generated by the same machine while spoofing only the last octet of its real public IP address. The payload consists of garbage memory data of what happens to be in the used memory space. The memory can also contain passwords and `ssh` private keys.

In the DNS Flood attack payloads, we see all distinguishable characteristics as well.

```
08:03:32.375073 IP xx.xx.223.184.21324 > 96.7.49.67.53: 45996 [1au] A? xx.xxxx20.com. (43)
        0x0000:  4500 0047 4c53 0000 4c11 64fd xxxx dfb8  E..GLS..L.d.....
        0x0010:  6007 3143 534c 0035 0033 8fea b3ac 0020  `.1CSL.5.3......
        0x0020:  0001 0000 0000 0001 02xx xx07 xxxx xxxx  .........xx.xxxx
        0x0030:  xx32 3003 636f 6d00 0001 0001 0000 2910  x20.com.......).
        0x0040:  0000 0000 0000 00                        .......
08:03:32.380600 IP xx.xx.223.184.21324 > 96.7.49.67.53: 45996 [1au] A? www.xxxxxx1.com. (44)
        0x0000:  4500 0048 4c53 0000 4c11 64fc xxxx dfb8  E..HLS..L.d.....
        0x0010:  6007 3143 534c 0035 0034 33a3 b3ac 0020  `.1CSL.5.43.....
        0x0020:  0001 0000 0000 0001 0377 7777 07xx xxxx  .........www.xxx
        0x0030:  xxxx xx31 0363 6f6d 0000 0100 0100 0029  xxx1.com.......)
        0x0040:  1000 0000 0000 0000                      ........
08:03:32.392463 IP xx.xx.223.184.21324 > 96.7.49.67.53: 45996 [1au] A? www.xxxxxxxxxxxxxx.com.
(51)
        0x0000:  4500 004f 4c53 0000 4c11 64f5 xxxx dfb8  E..OLS..L.d.....
        0x0010:  6007 3143 534c 0035 003b e168 b3ac 0020  `.1CSL.5.;.h....
        0x0020:  0001 0000 0000 0001 0377 7777 0exx xxxx  .........www.xxx
        0x0030:  xxxx xxxx xxxx xxxx xxxx xx03 636f 6d00  xxxxxxxxxxx.com.
        0x0040:  0001 0001 0000 2910 0000 0000 0000 00    ......).......
08:03:32.398063 IP xx.xx.223.184.21324 > 96.7.49.67.53: 45996 [1au] A? www.xxxxxx1.com. (44)
        0x0000:  4500 0048 4c53 0000 4c11 64fc xxxx dfb8  E..HLS..L.d.....
        0x0010:  6007 3143 534c 0035 0034 33a3 b3ac 0020  `.1CSL.5.43.....
        0x0020:  0001 0000 0000 0001 0377 7777 07xx xxxx  .........www.xxx
        0x0030:  xxxx xx31 0363 6f6d 0000 0100 0100 0029  xxx1.com.......)
        0x0040:  1000 0000 0000 0000                      ........
08:03:32.415895 IP xx.xx.223.184.21324 > 96.7.49.67.53: 45996 [1au] A? xx.xxxxx20.com. (43)
        0x0000:  4500 0047 4c53 0000 4c11 64fd xxxx dfb8  E..GLS..L.d.....
        0x0010:  6007 3143 534c 0035 0033 8ee0 b3ac 0020  `.1CSL.5.3......
        0x0020:  0001 0000 0000 0001 02xx xx07 xxxx xxxx  .........xx.xxxx
        0x0030:  xx32 3003 636f 6d00 0001 0001 0000 2910  x20.com.......).
        0x0040:  0000 0000 0000 00                        .......
```

Figure 4: Sample DNS Flood

As you can see, the observed DNS query flood is sourcing from the same source IP, with the same source port and the same query ID targeting multiple domains. The structure of all the packets might also look similar. This is because all of the target domains are passed from the C&C to the bots in a single command. From there the infected machines build the same template for all of the target domains.

We also see the Additional Record Count bit set across all packets and the DNSSEC options at the end.

Another not-so-important characteristic we noticed is that sometimes the DNS queries include the RD (Recursion Desired) flag on and sometimes it's off.

3.0 / RECOMMENDED DETECTION METHODS/ To detect this botnet in your network, one can look for the communications between bot and C2 using the following Snort rule:

```
alert TCP $HOME_NET any -> $EXTERNAL_NET any ( msg: "Xor-DDoS"; \
flow: established; \
content: "BB2FA36AAA9541F0BB2FA36AAA9541F0"; \
offset:32; depth: 64; \
classtype: trojan-activity; \
sid: 201500010; rev: 1;)
```

Figure 5: Snort rule matches on the XOR key string within the appropriate offsets in the initial registration between bot and C2

To detect infection of this malware on your hosts you can use the following Yara rule:

```
rule XorDDoSv1
{
        meta:
                author = "Akamai SIRT"
                description = "Rule to detect XorDDoS infection"

        strings:
                $st0 = "BB2FA36AAA9541F0"
                $st1 = "md5="
                $st2 = "denyip="
                $st3 = "filename="
                $st4 = "rmfile="
                $st5 = "exec_packet"
                $st6 = "build_iphdr"

        condition:
                all of them
}
```

Figure 6: Yara Rule

Here we used strings noticed in the binary itself:

string $st0 - *ASCII XOR key*
strings $st1 - *$st4 are parameters used by functionalities in the binary related to trojan activities.*
string $st5 - *function name to execute DDoS flood*
string $st6 - *function name to build IP header*

To match SYN Flood attack traffic generated by this botnet you can use the following `tcpdump` filters:

```
'ip[2:2] > 0x50' and 'tcp[13] & 2 !=0' and 'tcp[14:2] == 0xffff' and 'tcp[20:4] ==
0x020405b4' and 'tcp[24:4] == 0x01010402'
```

Figure 7: Example `tcpdump` filters for SYN flood attack traffic

This filter is composed of the following BPF filters:

```
'ip[2:2] > 0x50' - matches on total size field greater than 80 bytes
'tcp[13] & 2 !=0' - matches on SYN flags
'tcp[14:2] == 0xffff' - matches on TCP Window size 65535
'tcp[20:4] == 0x020405b4' - matches on static bytes used for TCP header options
'tcp[24:4] == 0x01010402' - matches on static bytes used for TCP header options
```

To match DNS Flood attack traffic generated by this botnet you can use the following `tcpdump` filters:

```
'udp[10:4] == 0x01200001' and 'udp[14:4] == 0x00000000' and 'udp[19] == 0x0001' and
'ip[((ip[2:2]) - 14):1] = 0x01'
```

Figure 8: Example `tcpdump` filters for DNS flood attack traffic

The above BPF filter is composed of:

```
'udp[10:4] == 0x01200001' - matches on the specific DNS fields from the outlisted
characteristics in Section 3
'udp[14:4] == 0x00000000' - matches on Answer Count and NS (Auth) Count fields
'udp[19] == 0x0001' - matches on Additional Record Count
'ip[((ip[2:2]) - 14):1] = 0x01' - matches on the DNS Query Type field for 'A' records
```

To remove the malware:

- Identify the main PID and the dropped files in `/boot` and `/etc/init.d`

```
root@ubuntu:/# ls -lha /boot | egrep " [a-z]{10}$"
-rwxr-x---  1 root root 606K Aug 19 11:42 utmhsahoca

root@ubuntu:/# ls -lha /etc/init.d/utmhsahoca
-rwxr-x--- 1 root root 28 Aug 19 11:42 /etc/init.d/utmhsahoca

root@ubuntu:/# find /proc/ -name exe 2> /dev/null | xargs -I{} ls -l {} 2> /dev/null | egrep
'/boot/[a-z]{10}$'
lrwxrwxrwx 1 root root 0 Aug 19 11:42 /proc/3746/exe -> /boot/utmhsahoca
```

Figure 9: Bash script used in the `/proc` file system process

The above bash script looks through all of the processes in the `/proc` file system and gets the path of their symlinked source finding all processes started from an executable in /boot with a 10-character name.

Here we have identified that `/boot/utmhsahoca` is currently running user PID **3746**.

The above script in more readable form and commented form:

```
# find all executing processes
1. find /proc/ -name exe 2> /dev/null

# get each processes symlinked destination
2. xargs -I{} ls -l {} 2> /dev/null

# find symlinks that match our '/boot/xxxxxxxxxx'
signature
3. egrep '/boot/[a-z]{10}$'
```

Figure 10: Additional Bash script

```
root@ubuntu:/boot# ps -eaf -u root | grep `date +%H:%M` | egrep -
v "ps|grep"
root      8318  2868  0 12:04 ?        00:00:00 ps -
ef
root      8321  2868  0 12:04 ?        00:00:00 cat
resolv.conf
root      8324  2868  0 12:04 ?        00:00:00 ls -
la
root      8326  2868  0 12:04 ?        00:00:00
id
root      8327  2868  0 12:04 ?        00:00:00 ls -
la
```

Figure 11: Identification for the supporting processes responsible for the persistence of the main process

The PIDs 8318, 8321, 8324, 8326, 8327 are short lived and their purpose is to only check if the main process is running. We are going to eliminate them by their start time, which will always be very close to the current time. If this command is run multiple times over the span of a minute the observed PIDs and associated fake commands will constantly change.

3.1 / KILLING THE MALICIOUS PROCESS /

```
root@ubuntu:/boot# ps -eaf | grep ^root | grep `date +%H:%M` | egrep -v "ps|grep" | awk '{print
$2}' | xargs -I{} kill -9 {}; kill -9 3746
```

Figure 12: Bash script in more readable form

```
# get processes
1. ps -eaf

# only get root users processes
2. grep ^root

# only keep processes started within the current minute
3. grep `date +%H:%M`

# remove grep and ps from list from the list
4. egrep -v "ps|grep"

# get the PID of the process
5. awk '{print $2}'

# force-kill the PID supplied for subprocesses
6. xargs -I{} kill -9 {};

# kill the main process
7. kill -9 3746
```

Figure 13: Additional Bash script

This script will get all processes running under the root user, it will then grab a subset of those processes that have spawned within the last minute, removing our own `ps` and `grep` from the list. It then extracts the PID for each process and kills it using -9, which cannot be blocked, prevented, or ignored. Once it has killed all of the watcher processes it kills off the master process, in this case PID **3746**.

Next we'll need to delete all of the malware files that have been dropped to disk.

```
root@ubuntu:/boot# rm -f /boot/utmhsahoca && rm -f /etc/init.d/utmhsahoca && /lib/udev/udev
```

Figure 14: Delete the dropped files in `/boot` and `/etc/init.d`

If you failed to kill the running instances of the malware, after step 4 the malware will drop a new copy of itself with a new 10-character randomized name in /boot and new startup script in `/etc/init.d`. If new files exist repeat steps 1-4 to ensure you've killed all running instances. You can also use the above YARA rule (Figure 6) on the whole file system to see if any copies of the malware are still present.

4.0 / CONCLUSION / Akamai's SIRT expects XOR DDoS activity to continue as attackers refine and perfect their methods.

This will likely result in a more diverse selection of DDoS attack types included in future versions of the malware. XOR DDoS malware is part of a wider trend of which companies must be aware: Attackers are targeting poorly configured and unmaintained Linux systems for use in botnets and DDoS campaigns.

A decade ago, Linux was seen as the more secure alternative to Windows environments, which suffered the lion's share of attacks at the time, and companies increasingly adopted Linux as part of their security-hardening efforts.

As the number of Linux environments has grown, the potential opportunity and rewards for criminals has also grown. Attackers will continue to evolve their tactics and tools and security professionals should continue to harden their Linux based systems accordingly.

Akamai
FASTER FORWARD

About Akamai Security Intelligence Response Team (SIRT) Focuses on mitigating malicious global cyber threats and vulnerabilities, the Akamai Security Intelligence Response Team (SIRT) conducts and shares digital forensics and post-event analysis with the security community to proactively protect against threats and attacks. As part of its mission, the Akamai SIRT maintains close contact with peer organizations around the world and trains Akamai's Professional Services and Customer Care teams to both recognize and counter attacks from a wide range of adversaries. The research performed by the Akamai SIRT is intended to help ensure Akamai's cloud security products are best of breed and can protect against any of the latest threats impacting the industry.

About Akamai® As the global leader in Content Delivery Network (CDN) services, Akamai makes the Internet fast, reliable and secure for its customers. The company's advanced webperformance, mobile performance, cloud security and media delivery solutions are revolutionizing how businesses optimize consumer, enterprise and entertainment experiences for any device, anywhere. To learn how Akamai solutions and its team of Internet experts are helping businesses move faster forward, please visit www.akamai.comor blogs.akamai.com, and follow @Akamai on Twitter

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 40 offices around the world. Our services and renowned customer care enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers and contact information for all locations are listed on www.akamai.com/locations