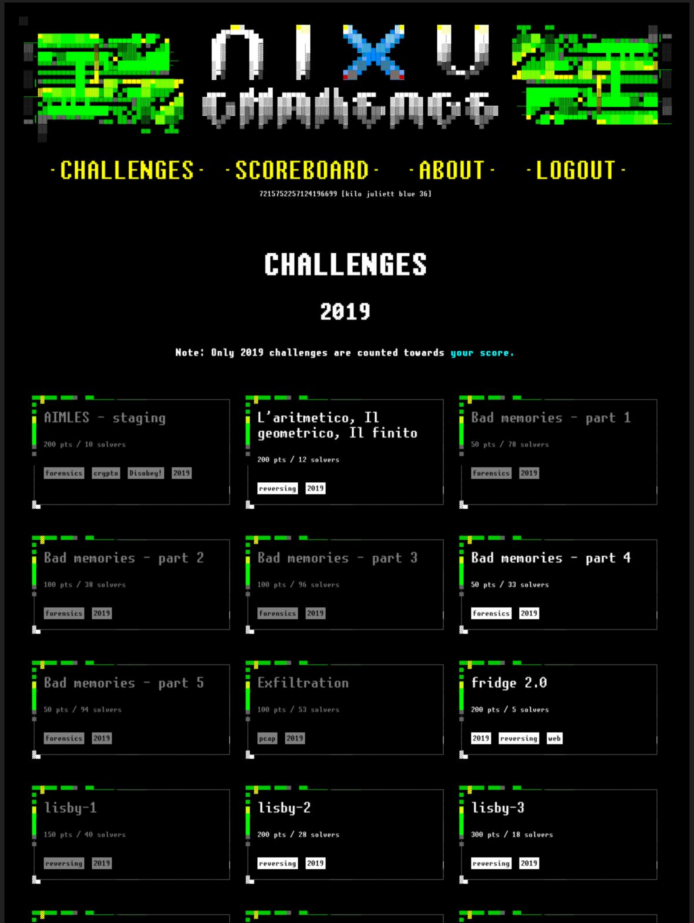
Solving the Nixu Challenges 2019

Final Presentation

Erik Sandberg
Patrick Richer St-Onge

Recap about the project

- Capture the Flag (CTF)
- Each challenge as a flag in the format NIXU{...}
- We try to solve the challenge (find the flag) and publish a write-up explaining how we did



7215752257124196699 [kilo juliett blue 36]

CHALLENGES

2019

Note: Only 2019 challenges are counted towards your score.

AIMLES - staging
200 pts / 10 solvers
forensics crypto Disobey! 2019

L'aritmetico, Il geometrico, Il finito
200 pts / 12 solvers
reversing 2019

Bad memories - part 1
50 pts / 78 solvers
forensics 2019

Bad memories - part 2
100 pts / 38 solvers
forensics 2019

Bad memories - part 3
100 pts / 96 solvers
forensics 2019

Bad memories - part 4
50 pts / 33 solvers
forensics 2019

Bad memories - part 5
50 pts / 94 solvers
forensics 2019

Exfiltration
100 pts / 53 solvers
pcap 2019

fridge 2.0
200 pts / 5 solvers
2019 reversing web

lisby-1
150 pts / 40 solvers
reversing 2019

lisby-2
200 pts / 28 solvers
reversing 2019

lisby-3
300 pts / 18 solvers
reversing 2019

Progress since last time

3 new challenges solved:

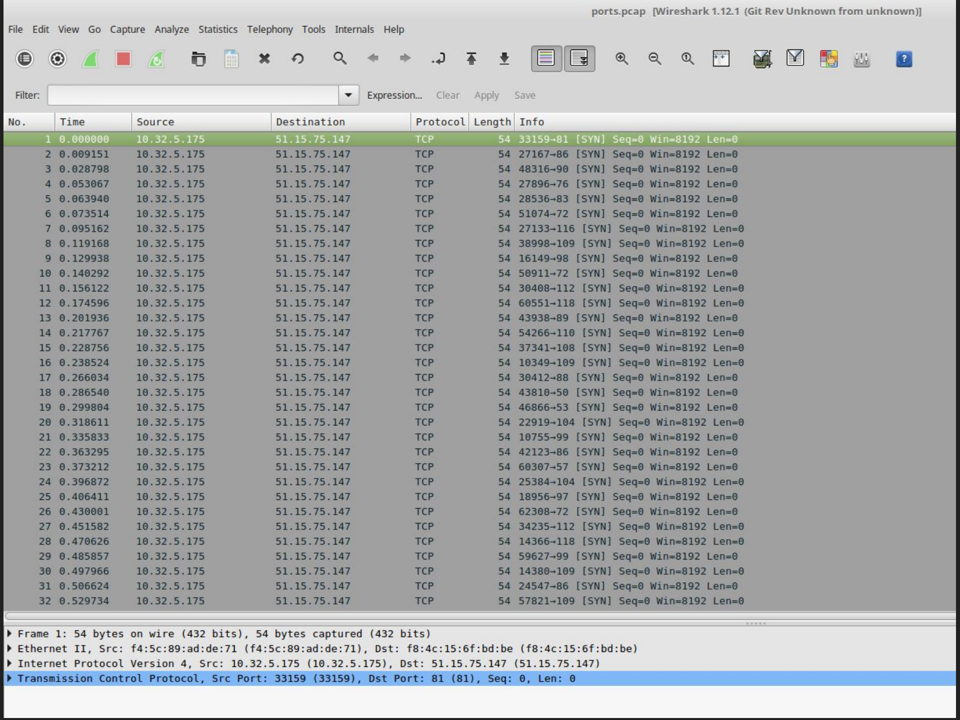
- AIMLES - staging
- Bad memories - part 2
- lisby-1

Current stats

- Solved 11/19 challenges
- Ranked #17 on the scoreboard

Ports

- Introctionary
- Wireshark, ASCII, Base64, ROT13
- Token encoded in port numbers
- Easily detected, disrupted by a basic firewall

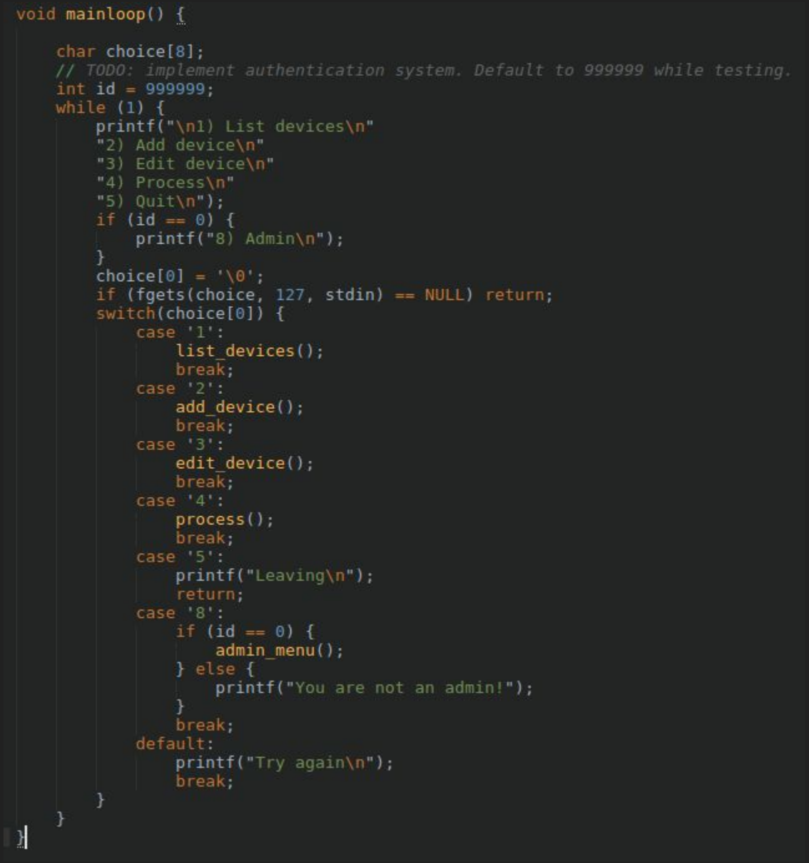

ports.pcap [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time      Source        Destination    Protocol Length Info
   1 0.000000  10.32.5.175   51.15.75.147   TCP          54 33159→81 [SYN] Seq=0 Win=8192 Len=0
   2 0.009151  10.32.5.175   51.15.75.147   TCP          54 27167→86 [SYN] Seq=0 Win=8192 Len=0
   3 0.028798  10.32.5.175   51.15.75.147   TCP          54 48316→90 [SYN] Seq=0 Win=8192 Len=0
   4 0.053067  10.32.5.175   51.15.75.147   TCP          54 27896→76 [SYN] Seq=0 Win=8192 Len=0
   5 0.063940  10.32.5.175   51.15.75.147   TCP          54 28536→83 [SYN] Seq=0 Win=8192 Len=0
   6 0.073514  10.32.5.175   51.15.75.147   TCP          54 51074→72 [SYN] Seq=0 Win=8192 Len=0
   7 0.095162  10.32.5.175   51.15.75.147   TCP          54 27133→116 [SYN] Seq=0 Win=8192 Len=0
   8 0.119168  10.32.5.175   51.15.75.147   TCP          54 38998→109 [SYN] Seq=0 Win=8192 Len=0
   9 0.129938  10.32.5.175   51.15.75.147   TCP          54 16149→98 [SYN] Seq=0 Win=8192 Len=0
  10 0.140292  10.32.5.175   51.15.75.147   TCP          54 50911→72 [SYN] Seq=0 Win=8192 Len=0
  11 0.156122  10.32.5.175   51.15.75.147   TCP          54 30408→112 [SYN] Seq=0 Win=8192 Len=0
  12 0.174596  10.32.5.175   51.15.75.147   TCP          54 60551→118 [SYN] Seq=0 Win=8192 Len=0
  13 0.201936  10.32.5.175   51.15.75.147   TCP          54 43938→89 [SYN] Seq=0 Win=8192 Len=0
  14 0.217767  10.32.5.175   51.15.75.147   TCP          54 54266→110 [SYN] Seq=0 Win=8192 Len=0
  15 0.228756  10.32.5.175   51.15.75.147   TCP          54 37341→108 [SYN] Seq=0 Win=8192 Len=0
  16 0.238524  10.32.5.175   51.15.75.147   TCP          54 10349→109 [SYN] Seq=0 Win=8192 Len=0
  17 0.266034  10.32.5.175   51.15.75.147   TCP          54 30412→88 [SYN] Seq=0 Win=8192 Len=0
  18 0.286540  10.32.5.175   51.15.75.147   TCP          54 43810→50 [SYN] Seq=0 Win=8192 Len=0
  19 0.299804  10.32.5.175   51.15.75.147   TCP          54 46866→53 [SYN] Seq=0 Win=8192 Len=0
  20 0.318611  10.32.5.175   51.15.75.147   TCP          54 22919→104 [SYN] Seq=0 Win=8192 Len=0
  21 0.335833  10.32.5.175   51.15.75.147   TCP          54 10755→99 [SYN] Seq=0 Win=8192 Len=0
  22 0.363295  10.32.5.175   51.15.75.147   TCP          54 42123→86 [SYN] Seq=0 Win=8192 Len=0
  23 0.373212  10.32.5.175   51.15.75.147   TCP          54 60307→57 [SYN] Seq=0 Win=8192 Len=0
  24 0.396872  10.32.5.175   51.15.75.147   TCP          54 25384→104 [SYN] Seq=0 Win=8192 Len=0
  25 0.406411  10.32.5.175   51.15.75.147   TCP          54 18956→97 [SYN] Seq=0 Win=8192 Len=0
  26 0.430001  10.32.5.175   51.15.75.147   TCP          54 62308→72 [SYN] Seq=0 Win=8192 Len=0
  27 0.451582  10.32.5.175   51.15.75.147   TCP          54 34235→112 [SYN] Seq=0 Win=8192 Len=0
  28 0.470626  10.32.5.175   51.15.75.147   TCP          54 14366→118 [SYN] Seq=0 Win=8192 Len=0
  29 0.485857  10.32.5.175   51.15.75.147   TCP          54 59627→99 [SYN] Seq=0 Win=8192 Len=0
  30 0.497966  10.32.5.175   51.15.75.147   TCP          54 14380→109 [SYN] Seq=0 Win=8192 Len=0
  31 0.506624  10.32.5.175   51.15.75.147   TCP          54 24547→86 [SYN] Seq=0 Win=8192 Len=0
  32 0.529734  10.32.5.175   51.15.75.147   TCP          54 57821→109 [SYN] Seq=0 Win=8192 Len=0
```

```
▸ Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
▸ Ethernet II, Src: f4:5c:89:ad:de:71 (f4:5c:89:ad:de:71), Dst: f8:4c:15:6f:bd:be (f8:4c:15:6f:bd:be)
▸ Internet Protocol Version 4, Src: 10.32.5.175 (10.32.5.175), Dst: 51.15.75.147 (51.15.75.147)
▸ Transmission Control Protocol, Src Port: 33159 (33159), Dst Port: 81 (81), Seq: 0, Len: 0
```

Device Control Pwnel

- Track of 2 challenges
- Overflow vulnerabilities in C code
- Misuse of secure function fgets
- Use of unsecure function strcpy
- Overflow bugs are known since 1972, still common



```
void mainloop() {

    char choice[8];
    // TODO: implement authentication system. Default to 999999 while testing.
    int id = 999999;
    while (1) {
        printf("\n1) List devices\n"
        "2) Add device\n"
        "3) Edit device\n"
        "4) Process\n"
        "5) Quit\n");
        if (id == 0) {
            printf("8) Admin\n");
        }
        choice[0] = '\0';
        if (fgets(choice, 127, stdin) == NULL) return;
        switch(choice[0]) {
            case '1':
                list_devices();
                break;
            case '2':
                add_device();
                break;
            case '3':
                edit_device();
                break;
            case '4':
                process();
                break;
            case '5':
                printf("Leaving\n");
                return;
            case '8':
                if (id == 0) {
                    admin_menu();
                } else {
                    printf("You are not an admin!");
                }
                break;
            default:
                printf("Try again\n");
                break;
        }
    }
}
```

Bad memories

- Track of 5 challenges, forensics
- Hands-on practice with Volatility (extraction of digital artifacts from memory)
- All the tools are there, just need to know where to look
  - Running processes
  - Keys in registry
  - Recently opened/modified/deleted files
  - And more...
- Useful for data recovery, digital forensics and malware analysis

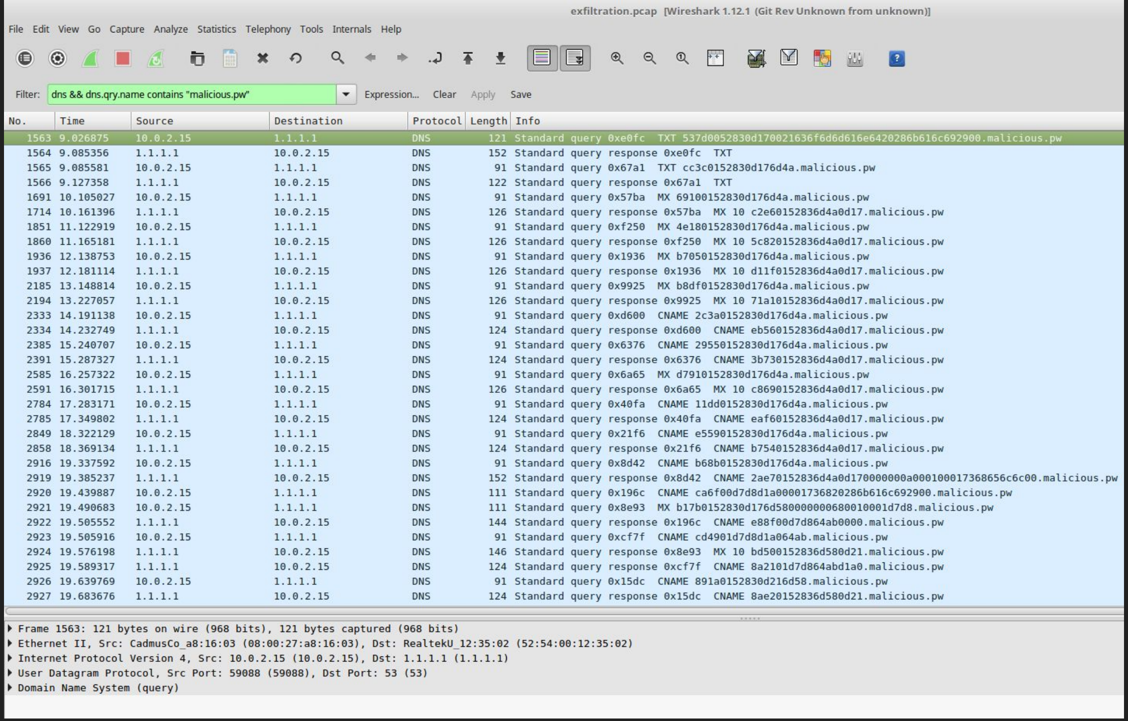
AIMLES

- Forensics, crypto
- Multiple steps challenge:
  - Finding hints in a network capture (pcap), extract ssh public keys
  - Exploit a flaw in key generation to recover ssh private keys
  - Using hints in pcap again, bruteforce TOTP secret
  - ssh using key + TOTP and get flag
- Somehow more similar to real world penetration testing (security audit)

TOTP = Time-based One-Time Password

Exfiltration

- Wireshark, DNS
- More realistic than Ports
- DNS rarely blocked
- Command-and-control with DNScat2


exfiltration.pcap [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: dns && dns.qry.name contains "malicious.pw"

```
No.   Time       Source      Destination  Protocol Length Info
 1563 9.026875   10.0.2.15   1.1.1.1      DNS         121 Standard query 0xe0fc  TXT 537d0052830d170021636f6d6d616e6420286b616c692900.malicious.pw
 1564 9.085356   1.1.1.1     10.0.2.15    DNS         152 Standard query response 0xe0fc  TXT
 1565 9.085581   10.0.2.15   1.1.1.1      DNS          91 Standard query 0x67a1  TXT cc3c0152830d176d4a.malicious.pw
 1566 9.127358   1.1.1.1     10.0.2.15    DNS         122 Standard query response 0x67a1  TXT
 1691 10.105027  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x57ba  MX 69100152830d176d4a.malicious.pw
 1714 10.161396  1.1.1.1     10.0.2.15    DNS         126 Standard query response 0x57ba  MX 10 c2e60152836d4a0d17.malicious.pw
 1851 11.122919  10.0.2.15   1.1.1.1      DNS          91 Standard query 0xf250  MX 4e180152830d176d4a.malicious.pw
 1860 11.165181  1.1.1.1     10.0.2.15    DNS         126 Standard query response 0xf250  MX 10 5c820152836d4a0d17.malicious.pw
 1936 12.138753  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x1936  MX b7050152830d176d4a.malicious.pw
 1937 12.181114  1.1.1.1     10.0.2.15    DNS         126 Standard query response 0x1936  MX 10 d11f0152836d4a0d17.malicious.pw
 2185 13.148814  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x9925  MX b8df0152830d176d4a.malicious.pw
 2194 13.227057  1.1.1.1     10.0.2.15    DNS         126 Standard query response 0x9925  MX 10 71a10152836d4a0d17.malicious.pw
 2333 14.191138  10.0.2.15   1.1.1.1      DNS          91 Standard query 0xd600  CNAME 2c3a0152830d176d4a.malicious.pw
 2334 14.232749  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0xd600  CNAME eb560152836d4a0d17.malicious.pw
 2385 15.240707  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x6376  CNAME 29550152830d176d4a.malicious.pw
 2391 15.287327  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0x6376  CNAME 3b730152836d4a0d17.malicious.pw
 2585 16.257322  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x6a65  MX d7910152830d176d4a.malicious.pw
 2591 16.301715  1.1.1.1     10.0.2.15    DNS         126 Standard query response 0x6a65  MX 10 c8690152836d4a0d17.malicious.pw
 2784 17.283171  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x40fa  CNAME 11dd0152830d176d4a.malicious.pw
 2785 17.349802  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0x40fa  CNAME eaf60152836d4a0d17.malicious.pw
 2849 18.322129  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x21f6  CNAME e5590152830d176d4a.malicious.pw
 2858 18.369134  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0x21f6  CNAME b7540152836d4a0d17.malicious.pw
 2916 19.337592  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x8d42  CNAME b68b0152830d176d4a.malicious.pw
 2919 19.385237  1.1.1.1     10.0.2.15    DNS         152 Standard query response 0x8d42  CNAME 2ae70152836d4a0d170000000a000100017368656c6c00.malicious.pw
 2920 19.439887  10.0.2.15   1.1.1.1      DNS         111 Standard query 0x196c  CNAME ca6f00d7d8d1a00001736820286b616c692900.malicious.pw
 2921 19.490683  10.0.2.15   1.1.1.1      DNS         111 Standard query 0x8e93  MX b17b0152830d176d580000000680010001d7d8.malicious.pw
 2922 19.505552  1.1.1.1     10.0.2.15    DNS         144 Standard query response 0x196c  CNAME e88f00d7d864ab0000.malicious.pw
 2923 19.505916  10.0.2.15   1.1.1.1      DNS          91 Standard query 0xcf7f  CNAME cd4901d7d8d1a064ab.malicious.pw
 2924 19.576198  1.1.1.1     10.0.2.15    DNS         146 Standard query response 0x8e93  MX 10 bd500152836d580d21.malicious.pw
 2925 19.589317  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0xcf7f  CNAME 8a2101d7d864abd1a0.malicious.pw
 2926 19.639769  10.0.2.15   1.1.1.1      DNS          91 Standard query 0x15dc  CNAME 891a0152830d216d58.malicious.pw
 2927 19.683676  1.1.1.1     10.0.2.15    DNS         124 Standard query response 0x15dc  CNAME 8ae20152836d580d21.malicious.pw
```

```
▸ Frame 1563: 121 bytes on wire (968 bits), 121 bytes captured (968 bits)
▸ Ethernet II, Src: CadmusCo_a8:16:03 (08:00:27:a8:16:03), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▸ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 1.1.1.1 (1.1.1.1)
▸ User Datagram Protocol, Src Port: 59088 (59088), Dst Port: 53 (53)
▸ Domain Name System (query)
```

Fridge 2.0

- Unfinished
- IoT, reverse engineering, unsafe crypto
  - Reverse firmware
  - Decrypt URL using key stored in firmware
  - ?
- Reasonably secure compare to its real life IoT counterparts
- 75 billion IoT devices by 2025
- Mirai botnet, over 1 terabit/s

ACME Order DB

- Web
- Able to bypass login exploiting a very simple flaw
  - But shows that authentication is commonly vulnerable in web app
- LDAP injection (similar to SQL injection)
  - Still exists today

lisby

- Track of 3 challenges, reversing
- Fictive/old computer architecture:
  - No toolchain (compiler, debugger, etc.)
  - Can’t use common tools (radare2, ghidra)
- Need to understand how a computer works
  - Combo: manual disassembling + scripting
- Useful for malware analysis and general reverse engineering skills

Conclusion

- Educational, fun, varied
- Wide knowledge-base is important for security
- Most attacks target basic vulnerabilities
- New insight into what knowledge actual security companies value

Questions?