https://portsentry.xyz

What is it?

Listen to Network Ports

Log Traffic

Execute Actions

Use-Case: Enumeration Prevention

- Listen to popular (unused) ports
  - telnet
  - ftp(s)
  - RDP
  - ldap
  - databases
- Ban connection attempts

Use-Case: Intrusion Detection

- Listen on common ports inside the organization
  - LAN
  - VPN
  - Management Network
  - WIFI
- On connection attempt: Trigger alarm

Use-Case: Statistics

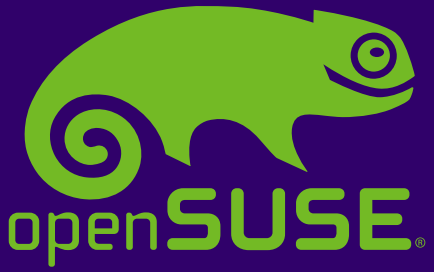
- Traffic Patterns
- Trends
- Anomalies


v0.01
v1.2 (abandoned)
Rebirth
v2.0 (soon)
1997
2003
2023
2025

v1.2 still available in...

debian

OpenBSD



gentoo linux™


ADD
FEATURES
BUGS
&
ISSUES

Portsentry 2.0

- Fixed Inconsistencies (DRY Violations)
- Fixed Resource leaks
- Fixed Race Conditions
- Fixed Input validation issues
- Code Consolidation
- **Optimizations**
- Deprecate / Warn dangerous functionallity
- Removed legacy code

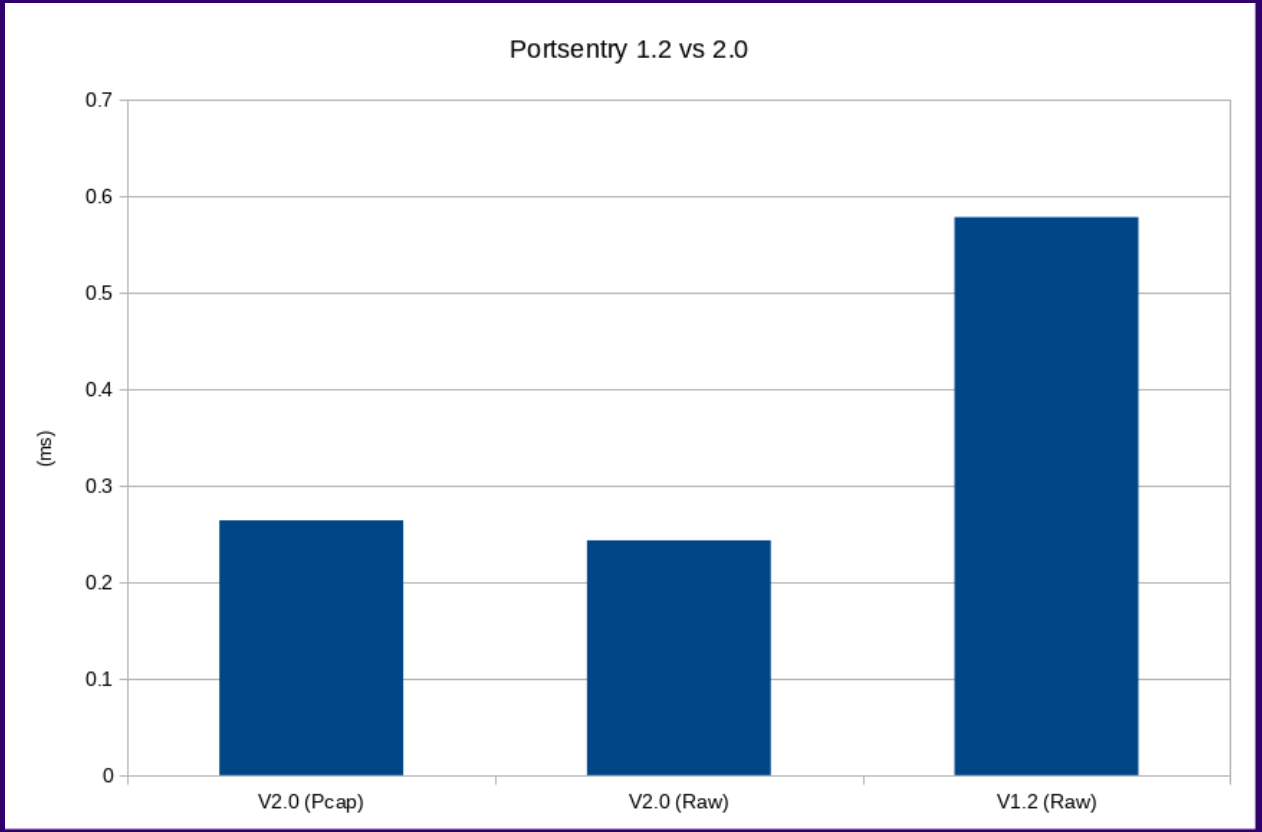
Portsentry 2.0

- Use CMake
- Linting, Formating
- Integration Testing
- Fuzzing
- SAST
- Documentation, HOWTO, Website
- Fine-grained configuration
- Better logging

Portsentry 2.0

- Libpcap support
- IPv6 Support
- Docker container, Docker Hub Registry
- Fail2ban integration

Unscientific profiling



Lessons Learned
Supporting multiple Un*xes

- Linux
- FreeBSD
- NetBSD
- OpenBSD

Lessons Learned X-Dev Unix

Linux != BSD != OpenBSD

Lessons Learned - Libc

NetBSD libc is missing getopt_long_only()

Lessons Learned - Compiler

BSD: clang

Linux: gcc

- Surprisingly compatible
- Some warning/error detection differences
- Clang behaviour changes more between versions

Lessons Learned - Headers

- BSD is more stingy with header #includes
- Must explicitly include more “base headers” such as types.h

Lessons Learned - Headers

Linux: struct iphdr

BSD: struct ip

But Linux has struct ip also :)

Also: Linux has two formats of:

- struct tcphdr
- struct udphdr
- Use the compatible one :)

Lessons Learned - AF_PACKET

socket(AF_PACKET, ...)

- *BSD will not sniff
- Linux will give you all packets

Lessons Learned – Kernel Events

Kernel Network Event Differences

Linux: socket(AF_NETLINK)

BSD: socket(PF_ROUTE)

*BSD: Different way of iterating messages

NetBSD:

- Rewamped at Kernel 8.0
- But kept backwards compatability
- But added some meta-data - help wanted by mystery solvers :)

```
#ifdef __NetBSD__
      unsigned char *bytes = (unsigned char *)sa;

      if (i == RTAX_IFA) {
        if (bytes[4] == 0x10 && bytes[5] == AF_INET) {
          ifa_addr_v4 = (struct in_addr *)(bytes + 8);
          sa = (struct sockaddr *)((char *)sa + 16);
        }
        else if (bytes[12] == 0x1c && bytes[13] == AF_INET6) {
          ifa_addr_v6 = (struct in6_addr *)(bytes + 20);
          sa = (struct sockaddr *)((char *)sa + 32);
        } else {
          sa = (struct sockaddr *)((char *)sa + 16);
        }
      } else {
        int len = (sa->sa_len > 0) ? sa->sa_len : 16;
        sa = (struct sockaddr *)((char *)sa + RT_ROUNDUP2(len, 4));
      }
#elif __FreeBSD__ || __OpenBSD__
      if (i == RTAX_IFA) {
        if (sa->sa_family == AF_INET) {
          ifa_addr_v4 = &((struct sockaddr_in *)sa)->sin_addr;
        } else if (sa->sa_family == AF_INET6) {
          ifa_addr_v6 = &((struct sockaddr_in6 *)sa)->sin6_addr;
        }
      }
#endif
#ifdef __FreeBSD__
      sa = (struct sockaddr *)((char *)sa + SA_SIZE(sa));
#elif __OpenBSD__
      sa = (struct sockaddr *)((char *)sa + ROUNDUP(sa->sa_len ? sa->sa_len : sizeof(struct sockaddr)));
#endif
```

Lessons Learned - Libpcap

- Linux, FreeBSD, NetBSD uses upstream libpcap
- OpenBSD forked libcap in 1996 and developed their own version… :o
- FreeBSD, NetBSD and OpenBSD differs from Linux on certain details

Lessons Learned – Dual Stack

IPv4/IPv6 dual stack

Linux	**NetBSD/ FreeBSD**	**OpenBSD**
Default Dual Stack	Setsockopt(...)	N/A

What you should do

- Continuously compile (and test) on all OS’es
  - That’s pretty much it :)

What’s Next

- v2.0 is right around the corner
- Prometheus integration
- Loki integration
- Multithreaded

Thank You

https://portsentry.xyz

portsentry@portsentry.xyz