OWASP NY/NJ Local Chapter

June 2007

An Educational Session on Embedded Systems and Their Associated Risks and Benefits.

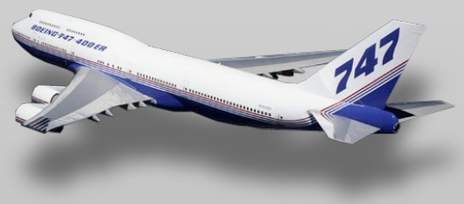
H|Y|D|R|A

Eric Ridvan Üner
CTO and Chief Scientist

This is how most people think of computers:



But there is an entirely different class:

What's the difference? Why do we care?

Exec Loop

µC/OS
VRTX

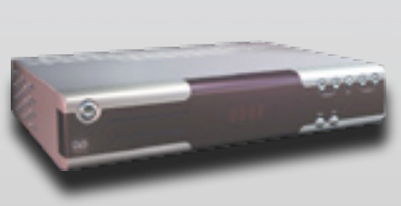
VxWorks
WinCE
QNX

RT Linux
Embedded Linux

Linux, BSD
Windows

Hard Real-time, Single Purpose

No Real-time, General Purpose



Embedded devices

These systems are often highly resilient to attacks, and in fact to any external programming. They can operate in environments where their failure could cause loss of life.

Network Appliances are on this end

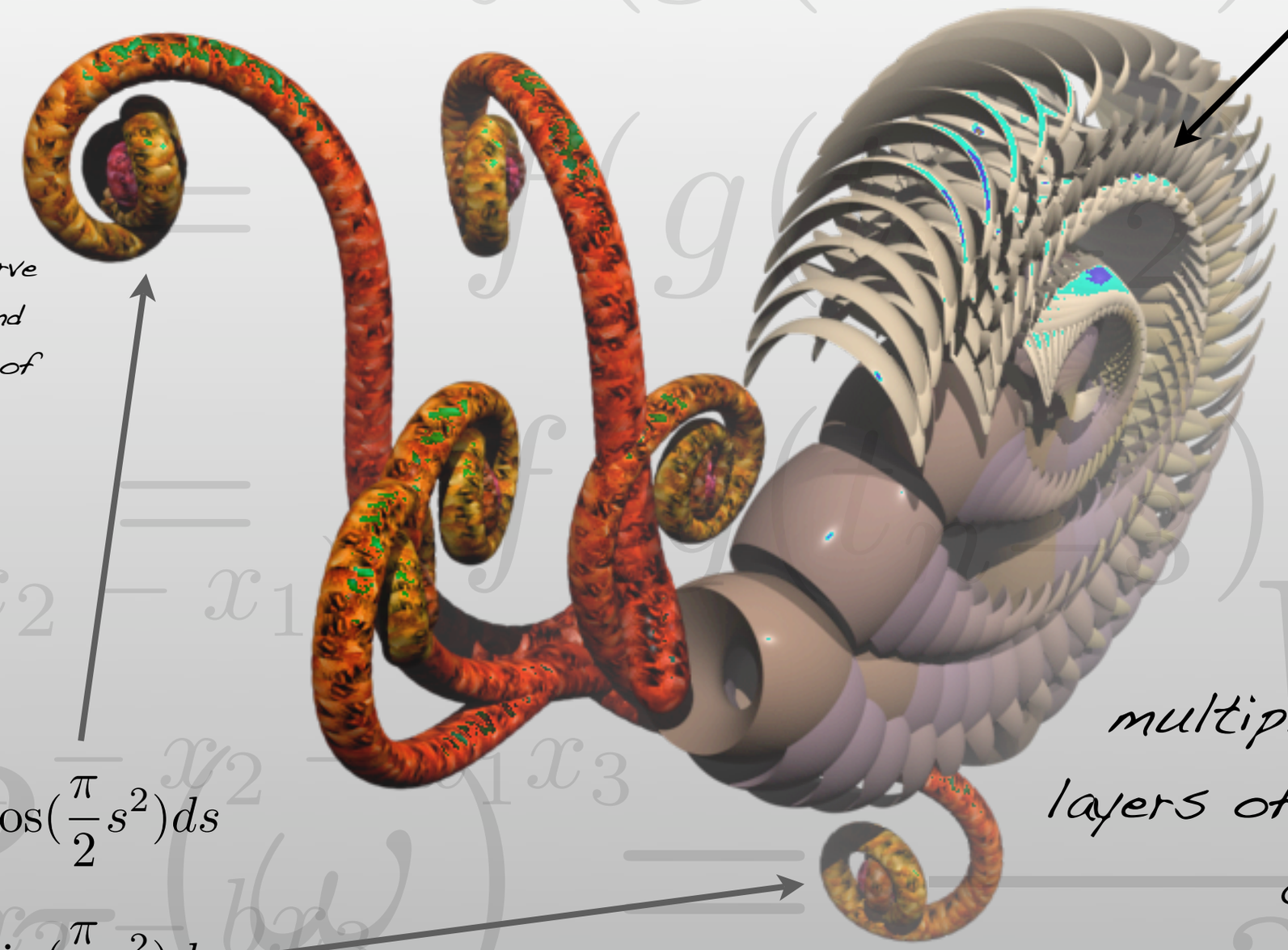
These devices use operating systems that are specifically designed to be open to external programming.

A helpful analogy...

Both sexual and asexual reproduction occur to balance diversity and birth rates

$$r(t) = e^{-\beta t}$$

specialized nerve from brain and ganglia at end of arms



multiple independent layers of protection and control

$$x = +\int_0^A \cos(\frac{\pi}{2}s^2)ds$$

$$y = -\int_0^A \sin(\frac{\pi}{2}s^2)ds$$

virology/parasites - see phylum Mesozoa

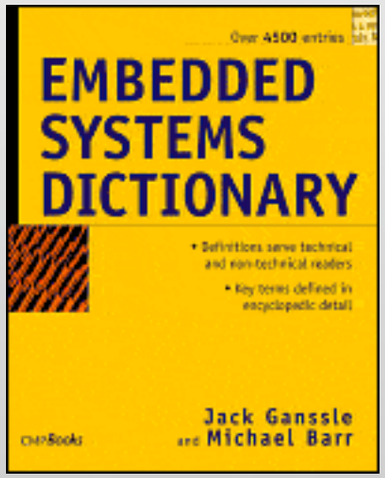
Consider Insects:

- Ubiquitous - there are at least 70 million of them per human
- Older and more robust than most life
- Specialists
- Our survival depends on them

Consider Embedded Systems:

- Ubiquitous - a modern car can have 100 processors
- Older and typically more robust
- Specialists
- Our survival depends on them

From the book "Embedded Systems Dictionary" by Jack Ganssle and Michael Barr



embedded system

n. A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a *dedicated function*. In some cases, embedded systems are part of a larger system or product, as in the case of an antilock braking system in a car.

OK, OK, I get it. They're everywhere. So what?

We care because as they enter the enterprise:

Since they *can* be more secure, we make false assumptions about their security posture.

Since they can be more secure, vendors like to say "embedded," and we again proceed from a false assumption.

For example:

Who would want to hack this?

Spammers, that's who.

Classic case of bad assumptions


Panasonic
BB-HCM331

Oops.


NetworkCamera
Pan / Tilt
Scan
Preset
Program
1 2 3 4
5 6 7 8
Brightness
- STD +
Resolution
640x480
320x240
160x120
Image Quality
Favor Clarity
Standard
Favor Motion
Image Size
x1.0 x1.5
Buffered Image


NetworkCamera
Pan / Tilt
Scan
Preset
1 2 3 4
5 6 7 8
Brightness
- STD +
Resolution
640x480
320x240
160x120
Image Quality
Favor Clarity
Standard
Favor Motion
Image Size
x1.0 x1.5
Buffered Image

How about these?

Java vulnerability allows arbitrary memory IO and execution

Symbian (not Windows)

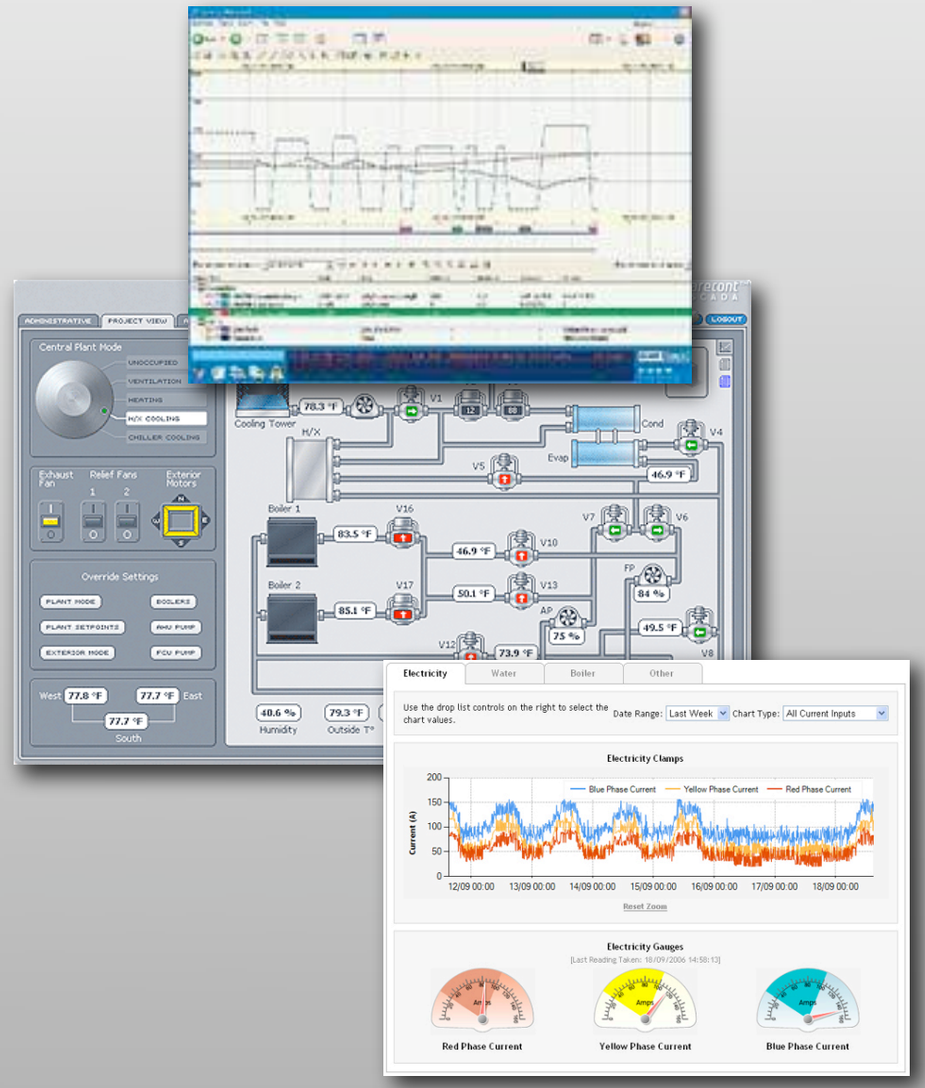
Could eventually change IMEI or worse on some phones

SCADA

Supervisory Control And Data Acquisition

+

=



It gets worse...

Nachi worm

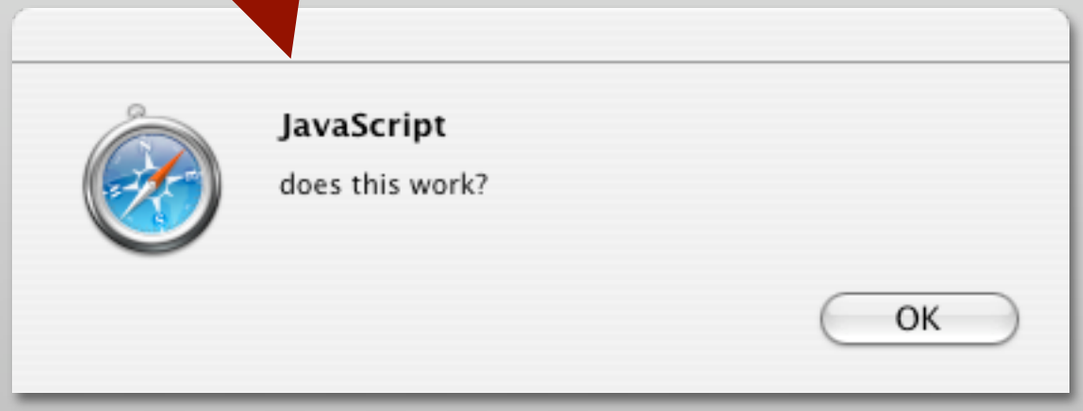
SANS: "Bad design decisions here: Running a single purpose appliance on a general purpose OS."

Appliances

- Popular firewalls are often BSD or Linux
- Many “appliances” are Windows
- For example in one of our tests...

Role Separation


```
[Apollo:~]
% telnet 10.0.1.1
Trying 10.0.1.1...
Password: *******
Ready:

1) Network Parameters
2) Access Parameters
3) Flash Programming
4) Exit

Ready: 1

1) Hostname
2) Primary IP Address
3) Web Administration Parameters
4) Exit

Ready: 1

Enter Hostname: <script>alert('does this work?');</script>Fake host

New Hostname: <script>alert('does this work?');</script>Fake host
OK ([y]/n)? y

1) Hostname
2) Primary IP Address
3) Web Administration Parameters
4) Exit

Ready:^c
[Apollo:~]
%
```



What do we do?

“Trust but Verify”

Treat them as enterprise devices

- Policy
- Procedure
- People

Policy

In general, apply enterprise policies. For example:

- iPods can be an attack vector
- Smartphones can be as bad as PC's
- Printer's Web server does not need to be publicly addressable
- Report all devices to IT

Procedure

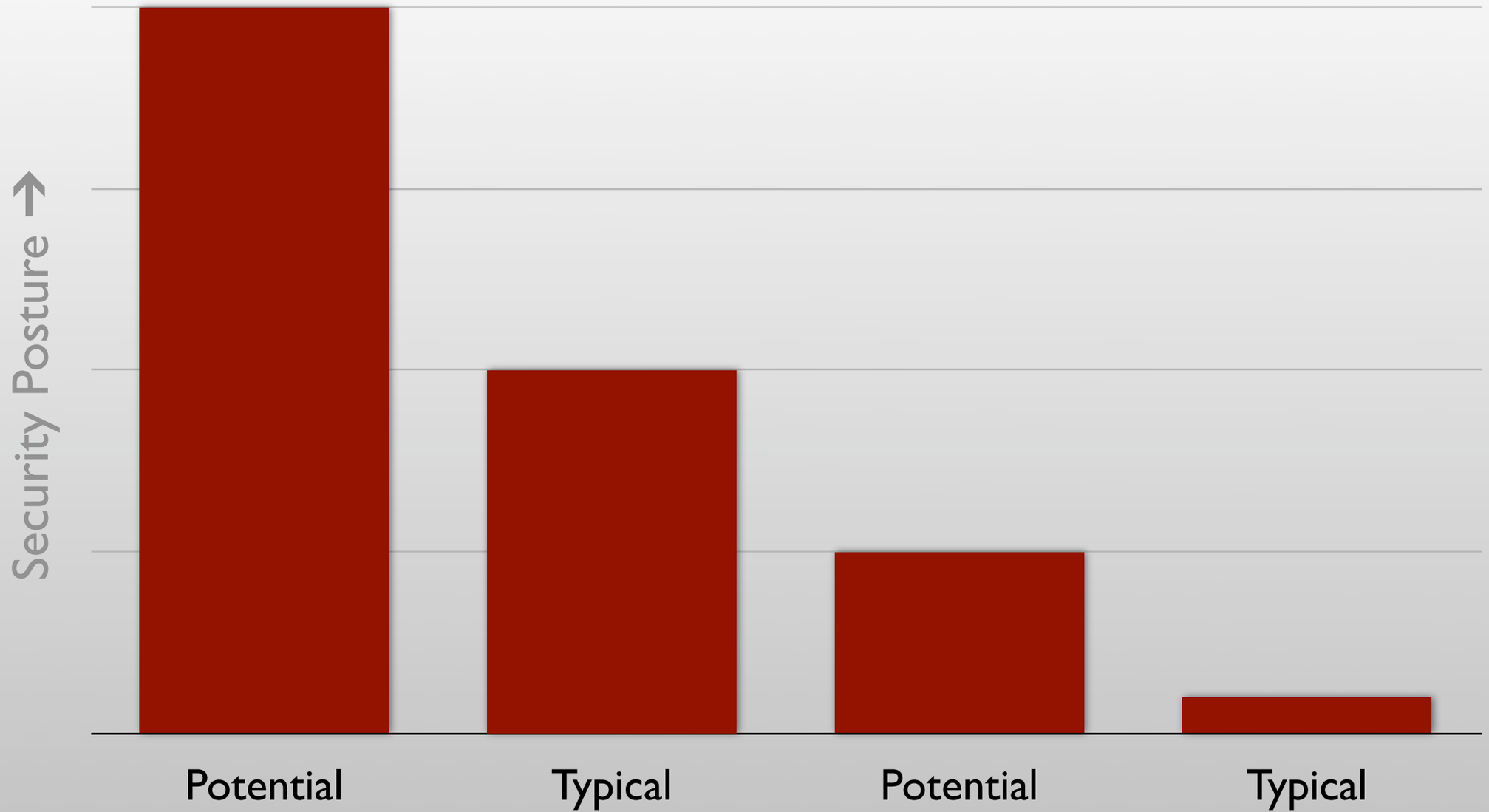
At a minimum:

- Use security scanners to scan devices
- Asses the threat of compromise
- Challenge vendors
- Audit

People

Educate them on the important lessons:

- If it's networked, or leaves your possession, it's a threat
- Make no assumptions
- Small computers are still computers
- Social engineering works, too



When are they less secure? When are they more?

Factors for Higher Risk

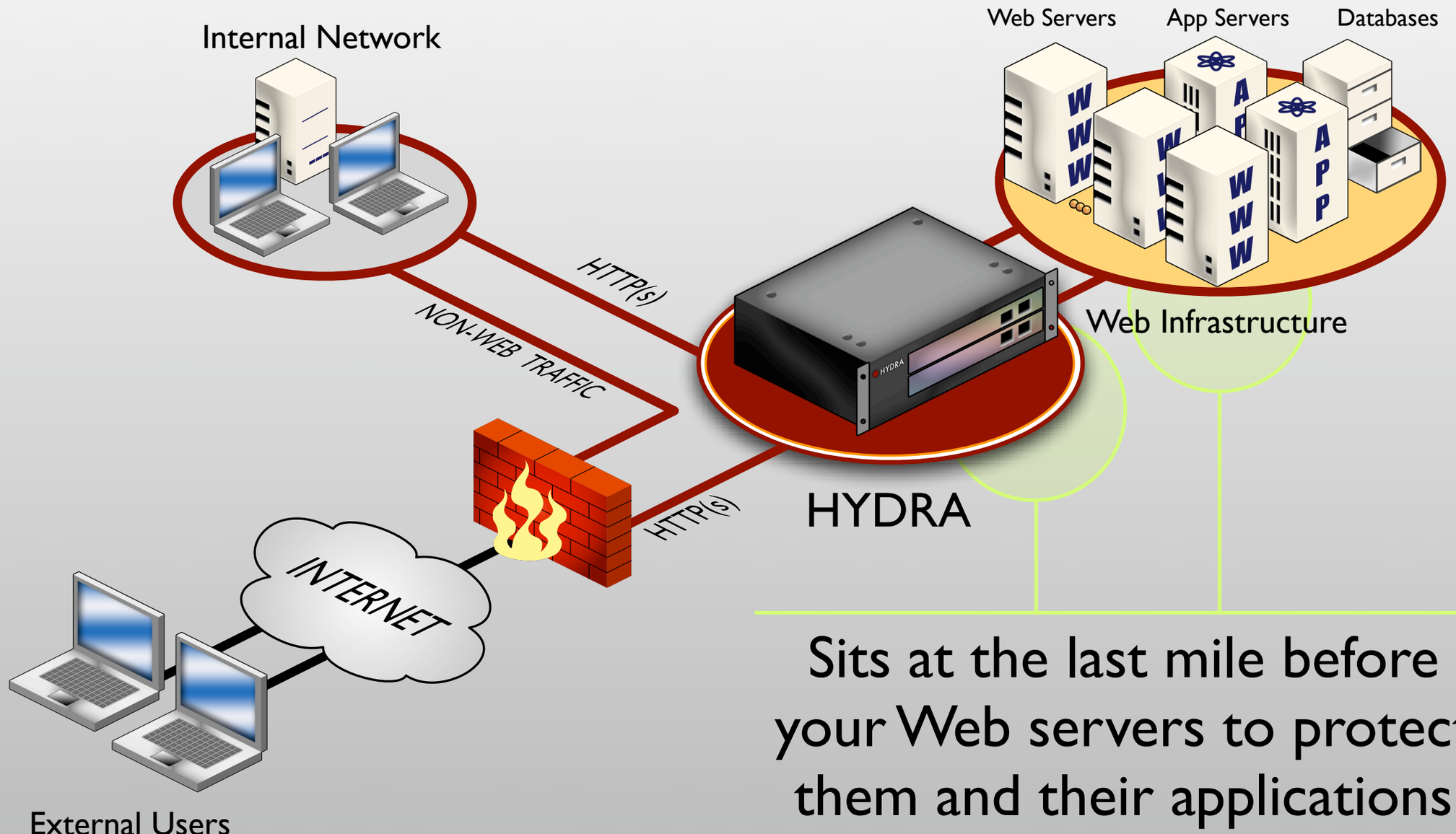
- These are red flags for further scrutiny, not indicators:
- Devices that are easily available and deployed in large quantities
- Offers a service outside it's domain (e.g. DSL router Web-based administration without Web security)
- Small devices with limited resources for checks
- Embedded + (any OS you heard of) e.g. Linux, BSD; rolled our own, [companyName]OS

Factors for Lower Risk

- Lack of connectivity
- Certifications like DO-178B Level A
- Look at other devices fielded with the same RTOS
- Was security part of the design?
- *Usage assumptions included hostile environment*

Shameless Plug

H|Y|D|R|A

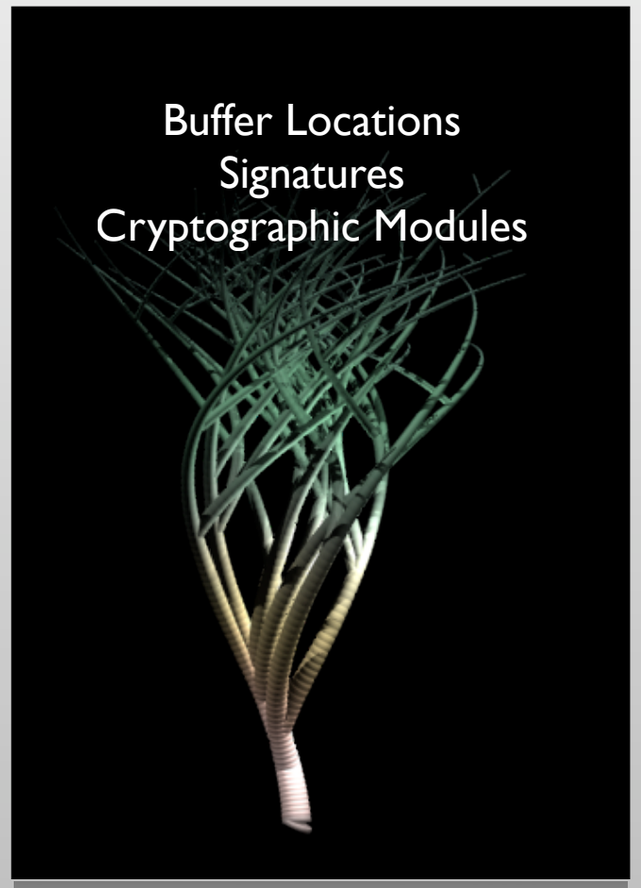


Sits at the last mile before your Web servers to protect them and their applications

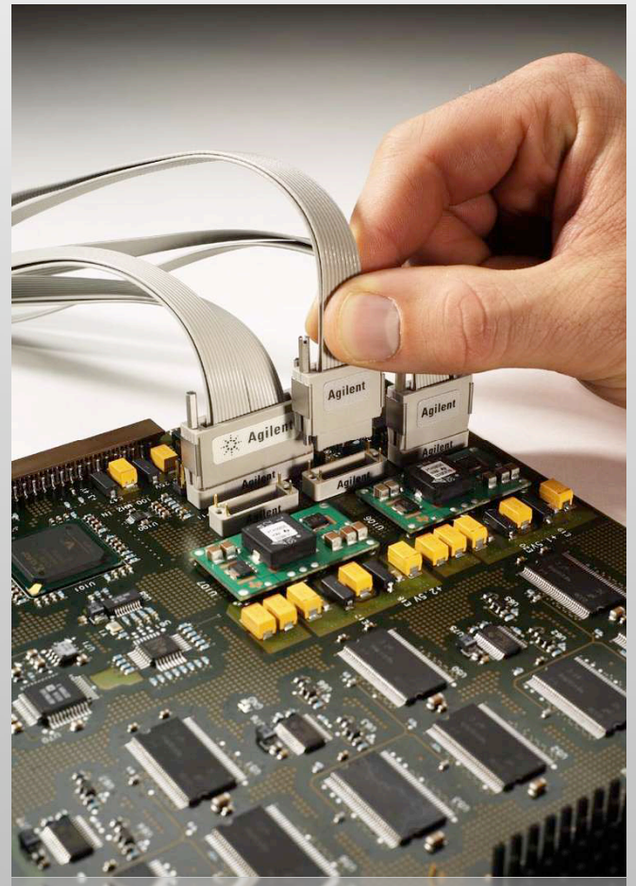
We needed a better security posture than what we were planning to protect.

Deep Design Requirements

Diversity



Buffers Kept From Probes



Phase Space Separation

- Looked at BSD
- Looked at Linux
- Looked at several RTOSs
- Settled on VRTX and INTEGRITY
- Good enough to fly these:

Good enough to protect Web servers

Big take-aways

- Identify an embedded system by it's characteristics (think insects)
- Watch out for "headless" systems passing themselves off as embedded
- Apply the same scrutiny to them that you do to enterprise systems
- Come see Sentinel Security Corporation

Thank You!
Questions?

Eric Üner

Uner@SentinelSecurity.US

More information on HYDRA is available at:

http://www.SentinelSecurity.US or
Sales@SentinelSecurity.US

Links to selected publications available at:

http://www.uner.com