Deployment of Cisco IronPort Web Security Appliance

Petr Růžička, CSE
CCIE #20166
Session Abstract

Web security proxy is a must for world full of Web 2.0 threats.

Focus of our session will be placement of Web Security Appliance (WSA) in the network, real world deployment scenarios, tips and tricks related to authentication of users, load balancing and scalability.

We would provide many answers to typical “how” and “why”. We assume knowledge of networking and basics of security concepts.
Agenda

1. Why
2. Testing
3. Deployment Options
4. Web Usage Controls
5. Q & A
Why
Web: Huge, Growing and Transient

Number of Webpages

1998: 28 Million webpages
2000: 1 Billion webpages
2008: 1 Trillion webpages

2005: Web 2.0 tipping point

Dynamic Web
User Generated & Web 2.0 Content

Static Web
Traditional Content Publishers
Legacy URL Filtering Focus

Huge
1 Trillion unique URLs

Growing
1 Billion new pages added every day

Transient
30% domain-level churn every year

Source: Multiple, including Cisco SIO, Google, Wikipedia
Traditional Firewall

1. ACL for L3/L4 filtering
   
   access-list inside permit udp DNS any eq 53
   access-list inside permit tcp LAN any eq 80
   access-list inside permit tcp LAN any eq 443
The New York Times: Victim of an advertiser attack!

1. Seemingly legitimate ad changed after NYT accepted it

2. 3 malicious redirections before malware reaches user

3. Ultimate destination: protection-check07.com

America’s Newspaper of Record
Testing
TESTING
I FIND YOUR LACK OF TESTS DISTURBING.
What we are after?

1. We are interested in requests per seconds
2. Secondary in throughput
3. Features of course are important
Following permance details are really VERY ROUGH estimates.
Suggested Positioning

S660 → 10,000 – 30,000 users
S360 → 1000 – 10,000 users
S160 → 0 – 1000 users
Why?

1. User’s browsing habits could be very different
2. Traffic profile could be different
3. Configuration and activated features could differ a lot
4. Some traffic could be denied
HTTP Performance
Requests/second & Throughput

- Key metric: Requests/second or Transactions/second
- The larger the object size, the higher the throughput
  \[ \text{Throughput (Mbps)} = \text{Object size (bytes)} \times \text{Request rate (req/s)} \]

  Throughput gets big… …as object size gets bigger… … even if the request rate stays constantly low.

Mbit/sec = Transactions Per Second (TPS) x Average HTTP object size (bytes) x bits in a byte (8) bits x in a megabit (1,000,000)

Rules of thumb:

- Each Request/second is approx. 80-90 Kbps of HTTP traffic
- Each Mbps of HTTP traffic translates to ~10 requests/second
- 100 Mbps of sustained HTTP traffic translates to ~1000 requests/second
Typical Traffic Profile

- HTTP: 70%
- HTTPS tunneled: 10%
- HTTPS decrypted: 10%
- FTP: 5%
- Other: 5%
Performance testing

1. Nothing beats real life traffic

2. Almost any tool that is used to test HTTP servers could be used here

3. Have a bunch of powerful servers ready

4. Test features relevant to your deployment
   Antimalware, HTTPs decryption, IP Spoofing…
Tools One Could Use

1. Wget
   
   ```
   wget -l 3 -r --proxy=on --delete-after http://guide.opendns.com/s?service=web&q=music&search_type=guide
   ```

   ```
   wget --proxy=on --delete-after --i list_of_sites.txt
   ```

2. Httperf
   
   ```
   http://sourceforge.net/projects/httperf/
   ```

3. Curl-loader
   
   ```
   http://curl-loader.sourceforge.net/
   ```

4. Pylot
   
   ```
   http://www.pylot.org/
   ```

5. Siege
   
   ```
   http://www.joedog.org/index/siege-home
   ```

6. Virus site of your choice :o)
Deployment
Web proxy modes

1. Explicit forward mode
   Use for evaluation
2. Transparent mode
3. Multiple upstream proxies mode
Proxy Selection Method

1. Deployment of Explicit Forward Proxy uniformly on many clients (Manual)
   - Avoid Configuration at the Desktop
   - Failover and/or Load Balancing
   - Performance

2. PAC file hosting
   - Locally on the desktop, on the server, or on the S-Series

3. WPAD
Explicit forward mode

1. Manual
2. PAC File
Creating the PAC file

1. Based on JavaScript
   Main function “FindProxyForURL(url, host)”
   12 PAC functions with return: DIRECT or PROXY

2. Different browsers behavior
   PAC file location
   IE limitations

3. Google PAC tester
   Test your PAC file if working properly
   http://code.google.com/p/pactester/

4. PAC file example
Creating the PAC file

```javascript
function FindProxyForURL(url, host) {
// If IP address is internal or hostname resolves to internal IP, send direct.
    var resolved_ip = dnsResolve(host);
    if (isInNet(resolved_ip, "10.0.0.0", "255.0.0.0") ||
        isInNet(resolved_ip, "172.16.0.0", "255.240.0.0") ||
        isInNet(resolved_ip, "192.168.0.0", "255.255.0.0") ||
        isInNet(resolved_ip, "127.0.0.0", "255.255.255.0"))
        return "DIRECT";

// Use a different proxy for each protocol.
    if (shExpMatch(url, "http:*")) return "PROXY proxy1.domain.com:3128";
    if (shExpMatch(url, "https:*")) return "PROXY proxy2.domain.com:3128";
    if (shExpMatch(url, "ftp:*")) return "PROXY proxy3.domain.com:3128";

// If I'm not in corporate network, send direct
    if (!isInNet(myIpAddress(),"144.254.0.0","255.255.0.0"))
        { return "DIRECT"; }

// URL based load balancing
    function FindProxyForURL(url,host){switch(URLHash(url)%2){
        case 1:  return "PROXY proxy1.domain.com:3128";
        default: return "PROXY proxy2.domain.com:3128";}}
    function URLHash(url) {server_name=url.split("/")[2]
        if(!server_name) {return url.length;}
        return server_name.length;}

// All other traffic uses below proxies, in fail-over order.
    return "PROXY 1.2.3.4:8080; PROXY 4.5.6.7:8080; DIRECT";
}
```
WPAD Web Proxy Autodiscovery Protocol (WPAD)

1. Using DHCP
   - Specification of PAC’s URL file using DHCP
   - In DHCP environment uses option 252 “auto-proxy-config”
   - Be aware of security when in use

2. Using DNS via A or CNAME
   
   host wpad
   
   PAC file should be then located on http://wpad.<domain>/wpad.dat

3. FF doesn’t support DHCP option, only DNS
Transparent Mode

1. User (browser, process...) doesn’t know it’s traffic is being proxied

2. Could be challenging sometimes, let’s say while having needs to authenticate users

3. Nothing needs to be done on users desktops
Transparent Proxy deployment

1. Configure transparent redirection
   PBR – L4 Switch, simple usage
   WCCP – WCCPv2 router, more complex, but more flexible

2. Configure the return method
   Layer 2 or GRE

3. Configure IP Spoofing or X-Forwarded-For Headers
   When upstream proxies require to identify the client
   Via Headers
   IP Spoofing in Forward mode

4. Configure Transparent Bypass List
   To exclude any special HTTP application servers or clients
**WCCP**

1. Identify traffic
   - Ingress redirection (preferred)
   - Egress redirection

2. WSA failure results in redistribution of load to remaining WSAs in 30 sec

3. If no remaining WSA’s service group is taken offline and packets are not redirected (fail-open)

```
FastEthernet0/0
ip wccp 80 redirect in
```
IP Spoofing Explained

1. Using proxy means connection to server would come from proxy address

2. Sometimes we need to identify client/subnet after proxy (think firewall, QoS, NAT, malicious requests detection)

3. Using spoofing we could, well, spoof 😊 source IP address

4. We would use original (clients) IP address
IP Spoofing in Forward Mode

Without redirection, the packets go straight back to the client!
Headers for Identification

1. **XFF (not sent by default)**
   
   X-Forwarded-For: client1, proxy1, proxy2

2. **Via (sent by default)**
   
   Used for identification of gateway or proxy server
   
   VIA: http protocol-version, host
From Client & To Client (request, answer)
From Proxy & To Proxy (request, answer)
ACE Loadbalancing Example

1. Using VIP (virtual IP address) for WSA cluster
2. ACE would provide HA and loadbalancing for both ESA and WSA
3. Could (and does) work for both transparent and explicit deployment
4. Using “IP Spoofing” gives us a option to identify traffic later in “green zone”
ACE Loadbalancing Bigger Picture
Cisco Web Usage Controls
Customer Problem

The Categorized Web
20% covered by URL lists
The Dark Web
80% of the web is uncategorized, highly dynamic or unreachable

- Dynamic content
- Password protected sites
- User generated content
- Short life sites

The Categorized Web
20% covered by URL lists
Introducing Cisco IronPort Web Usage Controls
A Spotlight for the Dark Web

1. Industry-leading URL database efficacy
   • 65 categories
   • Updated every 5 minutes
   • Powered by Cisco SIO

2. Real-time Dynamic Content Analysis Engine accurately identifies over 90% of Dark Web content in commonly blocked categories
Cisco Security Intelligence Operations (SIO)
Unmatched Visibility Drives Unparalleled Efficacy

Cisco SIO

Customer Administrators

Cisco IronPort Web Security Appliances on Customer Premises

Updates published every 5 minutes

URL Categorization Requests

Uncategorized URLs

Analysis and Processing

Master URL Database

Crawler Targeting

Traffic Data from Cisco IronPort Email Security Appliances, Cisco IPS, and Cisco ASA sensors

Uncategorized URLs

Manual Categorization

Web Crawlers

External Feeds

Crowd Sourcing
 Agenda

1. Why
2. Testing
3. Deployment Options
4. Web Usage Controls
5. Q & A
Registrujte se za Cisco Networkers
28-31. mart 2010. Bahrein

knowledge is power

January 25-28, 2010 • Barcelona, Spain

knowledge is power

March 28-31, 2010 • Bahrain