

```
0x8049500 <main+16>: mov     $0x0,0xfffff0(%ebp)
0x8049507 <main+23>: sub     $0x4,%esp
0x804950a <main+26>: push   $0x2ad
0x804950f <main+31>: push   $0x0
0x8049511 <main+33>: push   $0x805b2e0
0x8049516 <main+38>: call   0x8049338 <memset@plt>
0x804951b <main+43>: add     $0x10,%esp
0x804951e <main+46>: mov     %esi,%esi
0x8049520 <main+48>: sub     $0x4,%esp
0x8049523 <main+51>: push   $0x8055e40
0x8049528 <main+56>: pushl  0x0(%ebp)
0x804952b <main+59>: pushl  0x8(%ebp)
0x804952e <main+62>: call   0x8049338 <memset@plt>
0x8049533 <main+67>: add     $0x10,%esp
0x8049536 <main+70>: mov     %eax,%eax
0x8049538 <main+72>: mov     $0x0,0xfffff4(%ebp)
0x804953b <main+75>: jmp     0x8049544 <main+84>
0x804953f <main+79>: jne    0x8049544 <main+84>
0x8049541 <main+81>: jmp     0x80495c0 <main+208>
0x8049543 <main+83>: nop
0x8049544 <main+84>: Michael Sutton <msutton@idefense.com>
0x8049547 <main+87>: sub     $0x3f,%eax
0x804954a <main+90>: Pedram Amini <pamini@idefense.com>
0x8049550 <main+96>: cmpl   $0x37,0xfffffac0(%ebp)
0x8049557 <main+103>: ja     0x8049520 <main+48>
0x8049559 <main+105>: mov     0xfffffac0(%ebp),%edx
0x804955f <main+111>: mov     0x8055f30(,%edx,4),%eax
0x8049566 <main+118>: jmp     *%eax
0x8049568 <main+120>: mov     0x805a848,%eax
0x804956d <main+125>: mov     %eax,0x805b4dc
0x8049572 <main+130>: incl   0x805a524
0x8049578 <main+136>: jmp     0x8049520 <main+48>
0x804957a <main+138>: mov     %esi,%esi
0x804957c <main+140>: mov     0x805a848,%eax
0x8049581 <main+145>: mov     %eax,0xffffffe0(%ebp)
0x8049584 <main+148>: incl   0x805a528
0x804958a <main+154>: jmp     0x8049520 <main+48>
0x804958c <main+156>: mov     0x805a848,%eax
0x8049591 <main+161>: mov     %eax,0xffffffe4(%ebp)
0x8049594 <main+164>: jmp     0x8049520 <main+48>
0x8049596 <main+166>: mov     %esi,%esi
0x8049598 <main+168>: incl   0x805a52c
0x804959e <main+174>: jmp     0x8049520 <main+48>
0x80495a0 <main+176>: incl   0x805a530
0x80495a5 <main+183>: jmp     0x8049520 <main+48>
```

Hacking The Invisible Network

The Risks and Vulnerabilities Associated with Wireless Hotspots

Michael Sutton <msutton@idefense.com>

Pedram Amini <pamini@idefense.com>




```
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload_size = 2 * (datalen + 5);
#if defined DEBUG
printf("\nDEBUG>\ndata [%d]: ", datalen);
for (i = 0; i < datalen+5; i++)
printf("%x ", data[i]);
```

WISPs

Wireless Internet Service Providers

- aka Hotspots

What are they?

- Where are they?
 - Airports
 - Hotels
 - Retail stores
 - Coffee Shops
- Why go wireless?
 - Cost
 - Convenience



```
memcpy(chewycenter.payload, data, datalen + 5);
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload_size = 2 * (datalen + 5) + 10;
#ifdef DEBUG
printf("\nDEBUG>\ndata [%d]: ", datalen);
for (i = 0; i < datalen+5; i++)
printf("%x ", data[i]);
```

Industry

- Startups
 - Boingo
 - WayPort
 - NetNearU
 - HotSpotzz
 - Airpath Wireless
 - Surf and Sip
 - HereUAre
 - Deep Blue Wireless
 - Joltage (defunct)



100110010
 0110100101001010000101010100101010100101010010100101001010010100101001010010100101010001
 0101000010101010010010001010101000101011101011010100
 10101001011100101000101010110101010101001010010101001010100100101010010010100
 Copyright © 2003 iDEFENSE Inc.


```
memcpy(chewycenter.payload + dataalen + 5, data, dataalen + 5);
memcpy(chewycenter.payload + 2 * (dataalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (dataalen + 5) + 6 + 0)) = rdata;
chewycenter.payload_size = 2 * (dataalen + 5) + 10;
#if defined DEBUG
printf("\nDEBUG>\ndata %d: ", dataalen);
for (i = 0; i < dataalen+5; i++)
    printf("%x ", data[i]);
```

Provider Risks

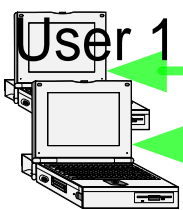
- Business risks
 - Financial loss
 - Launch pad for anonymous attacks
- Network level attacks
 - Privacy
 - Confidentiality
 - Data integrity
- Denial of service attacks
 - Availability




```
memcpy(chewycenter.payload, data, datalen + 5);
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long*)(chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload size = 2 * (datalen + 5) + 10;
#if defined(SECURITY)
printf("chewycenter.payload size = %d\n", chewycenter.payload size);
for (i = 0; i < datalen+5; i++)
printf("%x ", data[i]);
```

Security Implementations (cont'd)

User 1



Unplug and Play! WISE TECHNOLOGIES

ALREADY A MEMBER?

YES Please enter your username and password below to log in.

USERNAME

PASSWORD

Submit

NO Click the **Sign Up Now** button to select one of our **Daily** or **Monthly Summer Promo** accounts.

Sign Up Now

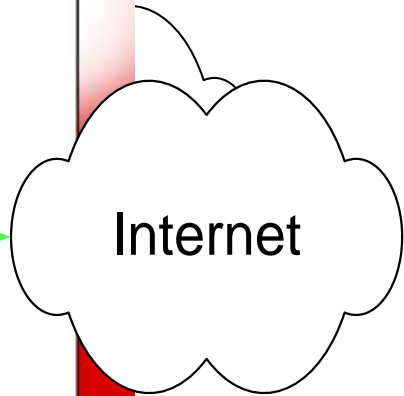
Click the **Temporary Access** button to surf the web **NOW** for only **\$0.20 per minute**.
That's 5 minutes for only \$1!

Temporary Access

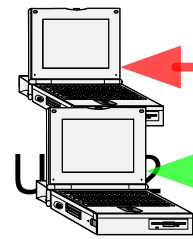
Powered by **nnu wireless**

8181 Professional Pl. Suite 200 Landover, MD. 20785 1-877-WISE-321

Access Point



Internet



User 2



```
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload size = 2 * (datalen + 5) + 10;
#if defined(SECURITY)
printf("chewycenter.payload size = %d\n", chewycenter.payload size);
for (i = 0; i < datalen+5; i++)
    printf("%x ", data[i]);
```

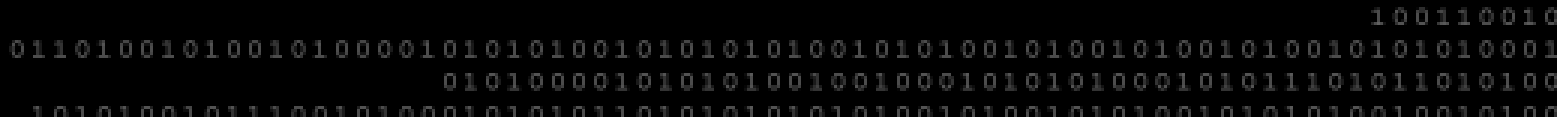
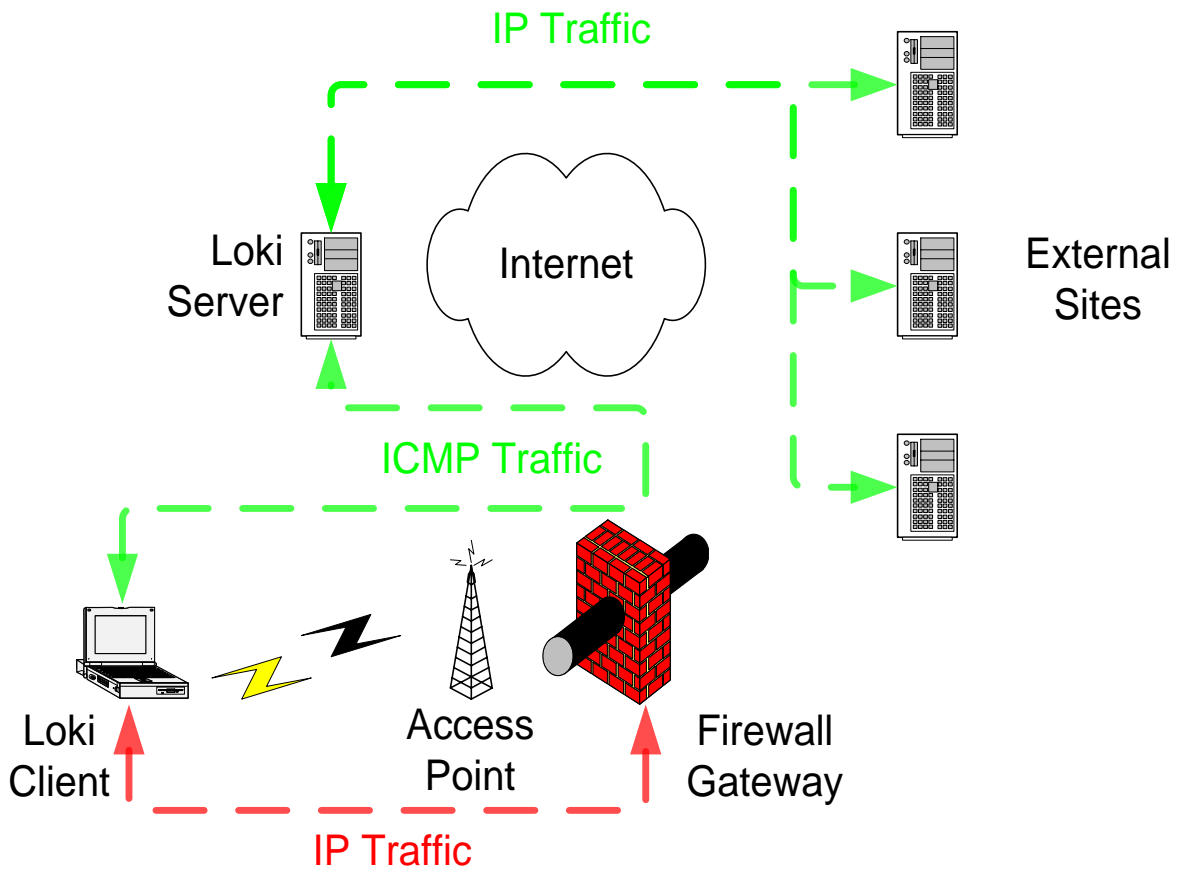
Security Implementations (cont'd)

- IP address filtering
 - Everyone
- MAC address filtering
 - T-Mobile
- IPsec VPN
 - Deep Blue Wireless
 - Optional
- DHCP lease expiration
 - ?



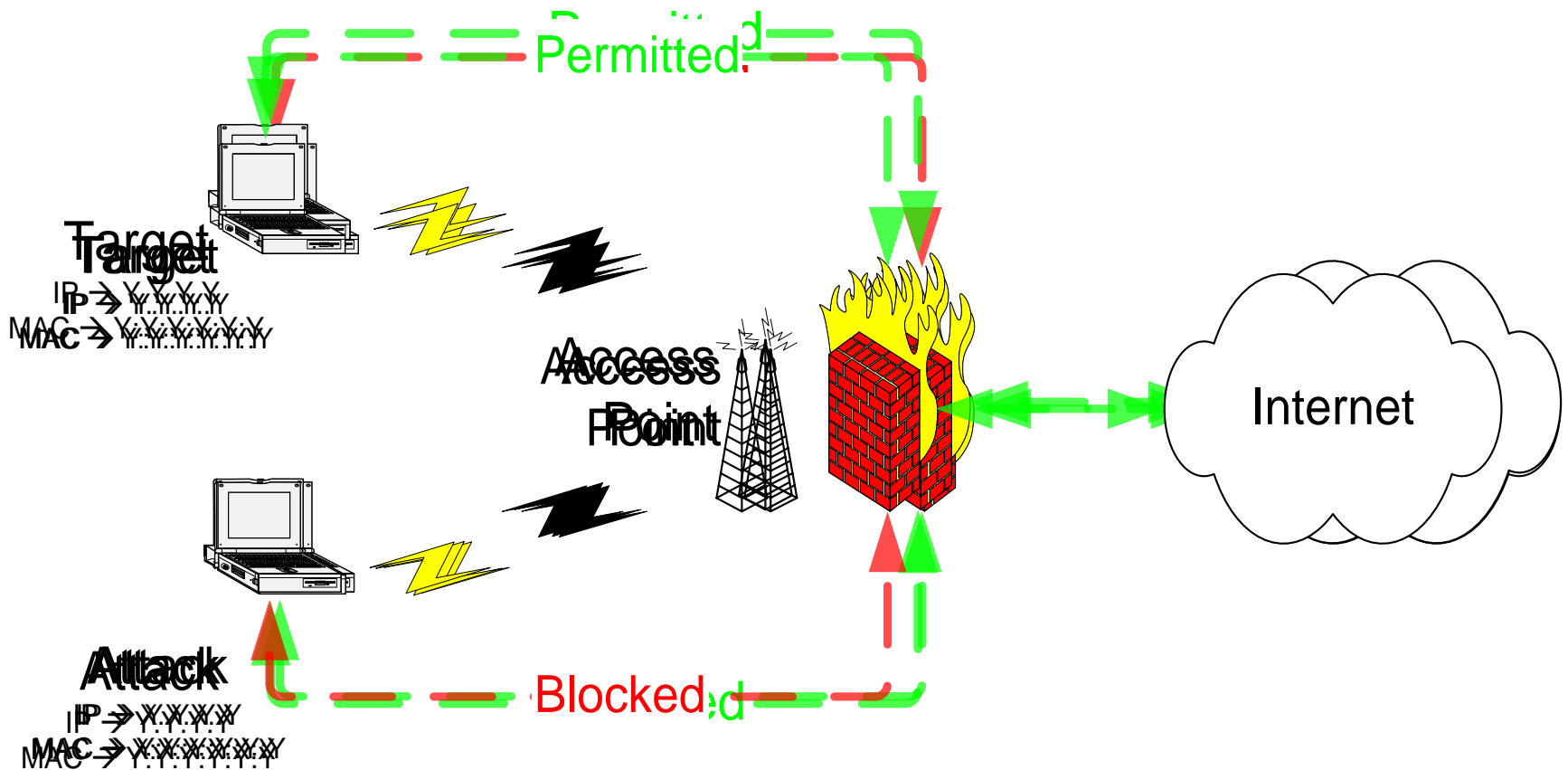

```
memcpy(chewycenter.payload, data, datalen + 5);
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload_size = 2 * (datalen + 5) + 10;
#if defined DEBUG
printf("\nDEBUG>\ndata [%d]: ", datalen);
for (i = 0; i < datalen+5; i++)
printf("%x ", data[i]);
```

Tunneling



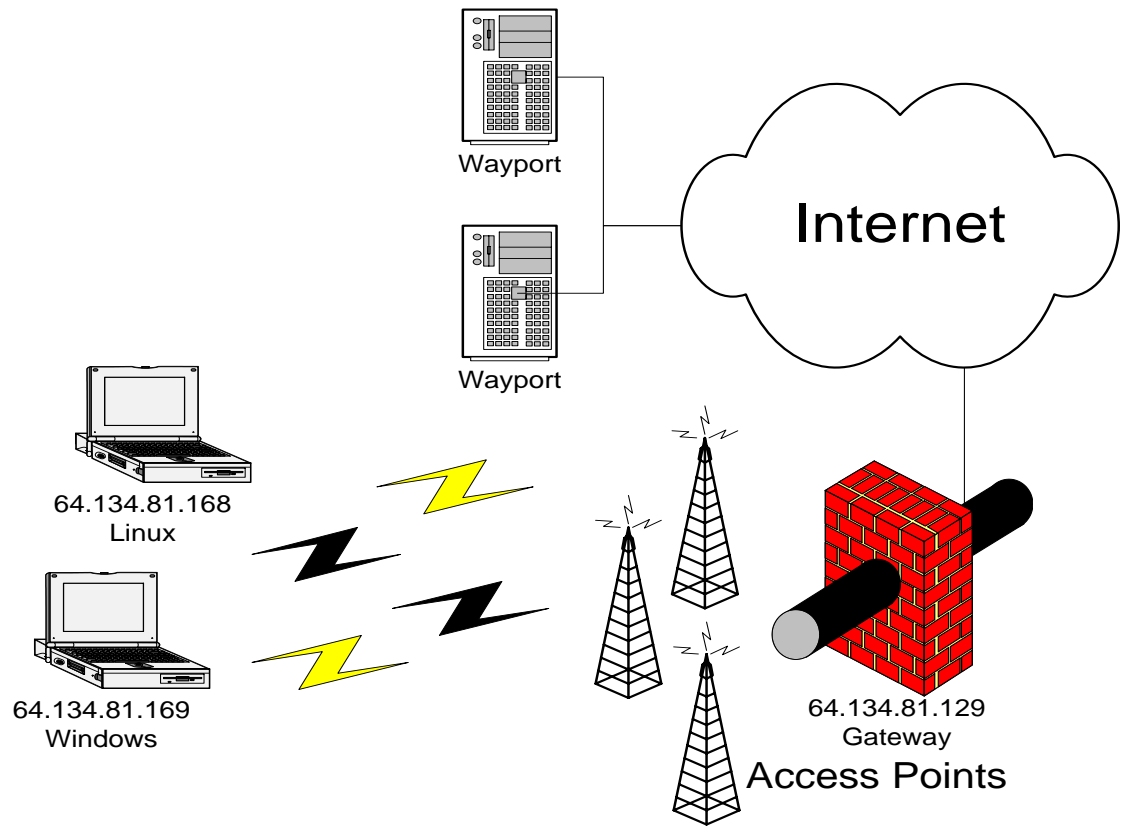
```
memcpy(chewycenter.payload + dataLen + 5, data, dataLen + 5);
memcpy(chewycenter.payload + 2 * (dataLen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (dataLen + 5) + 6 + 0)) = rdata;
chewycenter.payloadSize = 2 * (dataLen + 5) + 10;
#ifdef DEBUG
printf("\nDEBUG: dataLen: %d\n", dataLen);
for (i = 0; i < dataLen + 5; i++)
printf("%x ", data[i]);
```

Connection Hijacking




```
memcpy(chewycenter.payload, data, datalen + 5);
memcpy(chewycenter.payload + datalen + 5, data, datalen + 5);
memcpy(chewycenter.payload + 2 * (datalen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (datalen + 5) + 6 + 0)) = rdata;
chewycenter.payload_size = 2 * (datalen + 5) + 10;
#if defined DEBUG
printf("\nDEBUG>\ndata [0] = %d\n", data[0]);
for (i = 0; i < datalen+5; i++)
printf("%x ", data[i]);
```

WayPort Layout



```
memcpy(chewycenter.payload + dataLen + 5, data, dataLen + 5);
memcpy(chewycenter.payload + 2 * (dataLen + 5), "\x00\x00\x00\x00\x00\x04", 6);
*((u_long *) (chewycenter.payload + 2 * (dataLen + 5) + 6 + 0)) = rdata;
chewycenter.payload size = 2 * (dataLen + 5) + 10;
#if defined DEBUG
printf("\n\n%d\n", dataLen);
for (i = 0; i < dataLen+5; i++)
printf("%x ", data[i]);
```

Denial of Service Attacks

- Physical layer (1)
 - Interference
- Data layer (2)
 - ARP spoofing
- Network layer (3)
 - AirJack



