



**X-Force<sup>®</sup> Research and Development Newsletter**

April 2006

## Embedded Operating Systems

Each week, as we sit and watch the news, we hear about the latest security vulnerability or worm that places our computer at risk. The story has been repeating itself for over a decade — with only the names changing on a weekly basis. Worms such as Slammer or vulnerabilities like Microsoft's RPC DCOM have been consigned to the distant past and are now only referred to as case studies in computer security — after all, they are “ancient” and the flaws were fixed a long time ago.

Only the most naive organizations fail to ensure that all their desktop and server systems are patched on a regular basis. Whether they patch once a month or once a year, surely they do not have to worry about vulnerabilities that are several years old?

Unfortunately, and this may be a surprise to some people, there is no such thing as an old vulnerability or a dead worm.

While our desktop and server systems are typically in good shape, the same can not be said for all the other devices around us. Nowadays, if a piece of electrical equipment costs more than a few hundred dollars, you can pretty much guarantee that it can be connected to something else, has a management interface, can send and receive data, and almost certainly has its own embedded operating system. Think about it — just look at a multi-function laser printer fax machine.

What does this mean — should I be worried about these embedded operating systems? In a nutshell, — yes.

Let us examine that ubiquitous multi-function printer (MFP) — the bastion of productivity — accounting for a quarter of all the networked devices in a typical office. Most MFPs run a Linux or BSD operating system to provide basic services such as printing, faxing, scanning, document management and storage. With some higher end MFPs, you can scan a document and get it e-mailed back to you automatically. To provide these services, an MFP will typically have hundreds of gigabytes of data storage (both RAM and hard drive), a very fast processor, networking connections (both wired and wireless) and a fax/modem connection — basically becoming a server. One thing you will not find on it, however, is a CD-ROM drive. So if the MFP is running an embedded operating system, how do you harden it or even apply patches?



The same problem applies to almost all embedded operating systems. If you are lucky, the manufacturer makes available Flash-based updates to the devices, which may or may not include security fixes. Otherwise you may have to rely on a technician with a screwdriver.

The problem compounds itself if you are not aware that the device actually has an embedded operating system. For example, a well-known British retail banking organization discovered that it had a Blaster worm outbreak last year, due to vending machines connected to the corporate network so employees could use swipe cards to pay for snacks. Similarly, most “smart” buildings elevators contain networked services for remote management (e.g., between 6 a.m. and 10 a.m. the lift should wait at the 1st floor, and between 4 p.m. and 8 p.m. the lift should wait at the 5th floor). They are almost always forgotten about — thereby missing several years of patches — but are easily found by network-scanning worms or in penetration tests. Once they are compromised, beyond the annoyance of the elevators not working (which happens frequently — just ask any of the Las Vegas hotels that have ever hosted a hacker conference), peoples' lives may be in danger.

Perhaps the most disturbing threat from embedded operating systems can be found in hospitals and surgical wards. Most “cutting edge” equipment in these facilities will run very old mainstream operating systems, which are extensively networked but often missing several years' worth of security patches due to regulatory issues (which are in turn linked to insurance coverage). Speak with some of the senior IT security people within these hospitals and you will hear stories of how a 3-year-old-worm shut down vital equipment in the middle of surgery, killing the patient, or how wireless hackers interrupted their network, which caused a

heart monitor's emergency “flat-line” message to never reach the nurse. So much for a non-newsworthy computer virus being a nuisance — next time your life may be on the line!

What can be done to protect these embedded operating systems? This is an area core to ISS — it is our Virtual Patch™ protection strategy! Every single blocking signature in our products extends this protection, and is also one reason why we do not remove or depreciate our signatures — unlike several of our competitors. No one provides better embedded operating system protection than ISS.

-- Gunter Ollmann, Director of X-Force

## Inside the X-Force Engine

### XML Application Security

Have you heard the buzzwords Web 2.0, Web Services, and SOAP (Simple Object Access Protocol)? The common denominator in all this “pixie dust” is a W3C standard known as Extensible Markup Language (XML). Dozens of sub-standards such as XML Namespaces, XLink, XML Schema, XSLT, XQuery and RDF have been developed to boost interoperability and help application developers manipulate documents. You may be unfamiliar with this newer resource abuse attack vector and how the ISS Protocol Analysis Module (PAM) XML processor provides proactive protection against this threat.

XML was designed to provide a flexible way to structure application data that is programming-language neutral. The XML standard has no security functions such as authentication, integrity or privacy. Any service or product using XML must surround data with an application security component such as SSL/TLS or XML Security. The problem is compounded because many XML processor implementations have been created using different programming languages. Since each implementation is different, the security and Denial of Service (DoS) prevention features may vary. For instance, a Java-based processor may be immune to field overflow attacks, but a resource abuse attack (i.e., lots of memory allocations) is possible due to circular memory references and garbage collection.

An XML processor, depending on its level of robustness, may be susceptible to several different kinds of attacks such as resource abuse, entity expansion attacks, Document Type Declaration impersonation, SQL injection, character encoding evasion and buffer overflows. Many of these XML processors have to be explicitly configured to disable certain features such as processing Document Type Declarations, to reduce exploitation risk.

Resource abuse attacks are caused when attacker purposely transmits an overly long XML document, adds long tag names or creates long attribute lists. The example XML shown below is an abbreviated version of the exploit. In a real exploit, the sample data can be infinitely expanded and the victim would run out of network bandwidth, virtual memory or disk space before its Web service realized that document structure made no sense.

```
<?xml version="1.0"?>
<doc>
  <longStartTagABCD
    longAttributeNameABCD_1=
      "abcd&lt;&gt;&amp;&apos;&quot;"
    longAttributeNameABCD_2=
      "abcd&lt;&gt;&amp;&apos;&quot;"
    longAttributeNameABCD_3=
      "abcd&lt;&gt;&amp;&apos;&quot;"
  />
</doc>
```

Long XML documents can be problematic because the Web server acts as an application proxy; the Web administrator might not have configured any HTTP “POST” limitations. A Web service that is processing phone number verification requests could never successfully receive a 2MB document, but the entire document will be scanned before the processor declares defeat.

Long tag and attribute names can consume virtual memory beyond the string’s physical length. Depending upon the sophistication of the XML processor, enough virtual memory will be allocated to save each new name regardless of whether an identical value has been previously found. If the example document above had ten thousand `<longStartTagABCD>` tags, then a sophisticated processor would only allocate 16 bytes to store all the duplicate names; a less sophisticated processor would allocate 160KB. The system will quickly run out of memory — especially if several Web service requests are being handled concurrently.

Long attribute lists can consume exponential amounts of memory and induce memory fragmentation problems. Exponential amounts of memory are consumed when the XML processor chooses the wrong list storage data structure. Many XML processors read the entire list into memory. This leads to a subtle problem because the processor does not know the maximum number of elements beforehand; it must make a guess and adjust upwards as reading progresses. An attribute list length of 65 elements may have same storage requirements as 128 elements.

I am sure you will be glad to learn that PAM contains three signatures for supplementing an organization’s application security architecture. These signatures can filter documents based on document size, maximum tag or attribute name length, or maximum attribute list size.

- `XML_Document_Too_Large`. This signature detects when the content length of an XML document exceeds a tunable threshold, `pam.content.xml.limit`.

- XML\_Name\_Overflow. This signature detects when the number of bytes in an XML start tag, end tag, or attribute name exceeds a tunable threshold, pam.content.xml.name.limit.
- XML\_AttributeList\_DoS. This signature detects when the number of attribute name/value pairs exceeds a defined threshold, pam.content.xml.attribute.limit.

-- David Dennerline, X-Force Engineer

---

## Malcode Corner

### WMF vs. Zotob: The Threat Landscape is Changing

Zotob was a classic worm exploiting a vulnerability. The attacker would connect to port 445 on the victim. Firewalls are the classic defense — they typically block port 445 connections inbound to victims, and hence would block that threat. (It is not complete protection, of course, as we know worms often hitch a ride on laptops to get around firewalls. Let us just take for granted that a firewall will block an inbound connection).

In the case of Windows Metafile Format (WMF), attackers did not connect to victims; instead, victims connected to the attackers. There is no WMF software currently running that listens on a port — instead, the software is launched when you try to open a WMF file. To get a WMF file on your system, the attacker has to (typically) convince you to go to the malicious Web site and download it. This might be done through phishing e-mails, Google rank poisoning or some similar mechanism.

What do firewalls do in this case? Absolutely nothing. Firewalls stop inbound connections; they do not stop outbound connections. Even if the client machine goes through a proxy server to access the Internet, it can still download hostile WMF files.

So what can victims do to mitigate the WMF threat? Is antivirus the solution? Probably not.

Antivirus programs are useful at stopping well-known code from running on the system AFTER it has been exploited, but not stopping the exploitation itself. Furthermore, the hackers are getting better and better at making sure their code is not well known to the virus vendors.

What about IPS systems? Here is the dirty little secret of the IPS industry: most do not process responses from servers (for example, McAfee or Snort). Since 90 percent of historical attacks have been requests TO machines, but 90 percent of Internet traffic is in responses FROM machines, IPS systems have used the trick of ignoring the FROM traffic. They still catch most all attacks, but by only

processing 10 percent of the total network traffic. As more and more attacks are coming FROM servers, IPS products are becoming less and less useful.

How does ISS handle this? First, our Proventia® Network IPS does not use that trick of ignoring responses. Its gigabit sensors process the full gigabit traffic, both the TO and the FROM traffic. This is important when evaluating our products vs. our competitors: the evaluators test products by only sending attacks to victims, and do not evaluate attacks like WMF.

Second, we rely upon “defense in depth.” If our network IPS technology misses an attack, we still have host-based technology. Our buffer-overflow protection should prevent attacks like WMF from succeeding in the first place. However, if the exploit succeeds and malcode gets dropped on the victim, our Virus Prevention System (VPS) generic malcode detection should stop it from running — even if it is something that antivirus products will miss.

When ISS says *Ahead of the threat*, we do not mean simply that we were the first to have a signature for threats such as Zotob (though we were). It also means we are ahead of new threat classes such as “file format vulnerabilities” such as WMF. WMF highlighted the fact that our competitors really were not prepared for this new type of threat, but that ISS products are rock solid.

-- Robert Graham, Chief Scientist

---

## A Month in Review

Below is a brief summary of security content developed or processed in the month of March 2006.

### Malcode:

In March, the X-Force alpha-testing platform for the Virus Prevention System (codename Caffish) identified five unique 0-day malcode attacks — before any other antivirus vendor. These 0-day threats included a new Bagle mass-mailer/p2p worm, two variants of a Trojan with Rootkit capabilities and two new variants of W32/Brontok mass-mailer.

It is worth noting that, as part of the X-Force continued strengthening of ISS antivirus, anti-spyware and anti-malware protection, we added another 27,579 samples in the month to our lab for testing.

### Vulnerabilities:

Last month also saw 621 new vulnerabilities disclosed and processed by X-Force. This

represents a 62 percent increase over the same month last year.

	Reported					YTD
	Crit.	High	Med.	Low	Total	
Mar	4	97	378	142	621	1655

Of these vulnerabilities, 538 were remotely exploitable, 88 could only be exploited locally and 5 could potentially be exploited both remotely and locally.

Vulnerability Breakdown	
<b>Bypass Security</b>	<b>25</b>
An attacker can bypass security restrictions such as a firewall or proxy, an IDS system or a virus scanner.	
<b>Data Manipulation</b>	<b>106</b>
An attacker is able to manipulate data stored or used by the host associated with the service or application.	
<b>Denial of Service</b>	<b>74</b>
An attacker can crash or hang a service or system or take down a network.	
<b>File Manipulation</b>	<b>14</b>
An attacker can create, delete, read, modify or overwrite files.	
<b>Gain Access</b>	<b>269</b>
An attacker can obtain local and remote access. This also includes vulnerabilities by which an attacker can execute code or commands, because this usually allows the attacker to gain access to the system.	
<b>Gain Privileges</b>	<b>36</b>
Privileges can be gained on the local system only.	
<b>Obtain Information</b>	<b>86</b>
An attacker can obtain information such as file and path names, source code, passwords or server configuration details.	
<b>YTD Total</b>	<b>1655</b>

Right-hand column represents unique vulnerability count

Top 5 Vulnerable Vendors	
Microsoft	18
Mac OSX	14
Linux Kernel	14
HP	6
Mantis	6

Right-hand column represents unique vulnerability count

### X-Force Security Content:

March saw X-Force add 27 new events for various vulnerabilities, deliver two Level-0 X-Press Updates (XPUs), and add 69 new blocking responses

### Trust X-Force – 1028 default blocks

Last month ISS reached a major milestone in its IPS history. *PAM now blocks over 1000 vulnerabilities by default — that is 46 percent of the decodes covered in PAM!*

Just over a year ago, X-Force started an initiative to block high-priority vulnerabilities with PAM. With each XPU, X-Force engineers conducted code reviews and collated data from Managed Security Services (MSS) and PAM Alpha sensors to determine exactly which decodes could safely be implemented for blocking.

Any decode that could possibly produce legitimate false positives was not blocked, and any false positives that were discovered during the reviews have been fixed.

After each monthly XPU, X-Force reviews the data for decodes contained in the last scheduled XPU to determine what decodes could be safely blocked. Any blocked decodes that have exhibited false positives have their blocking removed, and blocking is not re-enabled until the false positive has been removed and the decode has been back in the field without any further false positives for at least one month.

You may review the XPU readmes to see what blocking changes have been made with each XPU and check out X-Force on the ISS Intranet (<https://issintranet.iss.net/xforce/PAMblocklist.htm>) to see the full list of blocks.

1000 Blocks — That is a lot of work, from a lot of engineers, over several years. Congratulations to all those engineers for helping to take Proventia into a realm unmatched by any competitor!

### Extended Protocol Support:

During March, X-Force engineers extended fourteen existing PAM protocol parsers. This included the following protocols:

- HTTP
- JavaScript
- MSRPC

- LPR
- DNS
- Brightstor
- TCP
- SSL
- RPC
- LDAP
- ICMP
- FTP
- IMAP
- VERITAS

**New Vulnerability Discovery Algorithms:**

X-Force added the following security coverage to ISS products:

- JavaScript\_Shellcode\_Detected
- HTTP\_Real player\_Chunked\_Overflow
- JavaScript\_NOOP\_Sled
- HTML\_Mshtml\_Overflow
- LPD\_Cachefsd\_Overf
- RLPR\_Format\_String
- BrightStor\_Discovery\_UDP\_Overf
- HTTP\_Linksys\_ApplyCgi\_BO
- SIP\_Proxy\_Overf
- Ace\_Filename\_Overf
- HTTP\_Winamp\_Detected
- Tagged\_Packet
- HTTP\_Media\_Player\_Detected

RealMedia\_Detected

- RTSP\_Detected
- LDAP\_BER\_Sequence\_Dos
- WindowsMedia\_Detected
- HTTP\_Slingbox\_Detected
- MOV\_Detected
- RPC\_Mountd\_Packet\_Length\_Corrupt
- StreamingMedia\_Detected
- HTTP\_KCeasy
- Gift\_Download
- LPD\_ControlFile\_Overflow
- LPD\_Unix\_Shell\_Command
- SQL\_XP\_CMDSHHELL\_Detected
- EPolicy\_Orchestrator\_POST\_Overf

**New and Extended Content Support:**

X-Force engineers extended seven existing PAM content parsers, including the following content formats:

- JavaScript
- HTML
- XML
- SDP
- Jpeg
- Tiff
- Text

**Copyright© 2006 Internet Security Systems, Inc. All rights reserved worldwide.**

Internet Security Systems, the Internet Security Systems logo, Proventia, the Proventia logo, SiteProtector and Ahead of the Threat are trademarks or registered trademarks of Internet Security Systems, Inc. Other marks and trade names mentioned are the property of their owners, as indicated. All marks are the property of their respective owner and used in an editorial context without intent of infringement. Specifications and content are subject to change without notice.