



X-Force[®] Research and Development Newsletter

May 2006

RFID: Security Threat?

Media interest in Radio Frequency Identification (RFID) has increased substantially in recent months, and you may have been quizzed about the security implications of the technology.

RFID technology is based on attaching a chip with a small antenna (referred to as an RFID Tag) to an object that will emit a unique signal response (the identifier) when a specific radio frequency is directed at it. The most common forms you will encounter are the passive RFID tags (having no internal power) attached to books and other merchandise (usually stuck to the packaging) to prevent theft.

RFID tags have been in use since the 1980s and are increasingly used for all manner of jobs. The most common usage is as a form of inventory control and tracking. As an example, in the early 1990s most clothing



rental companies (think in terms of black-tie events and weddings) started sewing RFID tags into their garments for inventory management. The tags themselves are encased in a special plastic that can survive repeated exposure to the chemicals used in dry-cleaning, are uniquely linked to each individual item and have their details stored in a database system. Why not use a barcode instead and just scan it? Imagine that a truck pulls into a warehouse with 5,000 items on board — scanning each barcode would take a long time. Instead, by using RFID tags, the hauler just throws the boxes of items down a chute and each tag is read automatically — without even having to open the boxes.

So how does this warrant an identity threat? Using clothing as an example, have you ever bought an expensive suit from a store, and then returned to the store several months later to be greeted personally, or wondered how the salesman knew your sizes perfectly? Yes, you guessed it: a lot of expensive suits have embedded RFID tags sewn into them (often in the collar or lapel) for inventory control and to combat “stock shrinkage” (i.e., theft). They are supposed to be disabled after the sale, but are often forgotten about.

Today’s RFID tags can be very small (the smallest are 0.15mm x 0.15mm) and may be implanted or embedded in anything. They typically carry no more

than 2KB of data and some may even contain writable memory; but the most common mass-produced versions normally possess only memory enough for a read-only 96-bit serial number.

RFID security threats can be broken into two categories — threats that use the data, and threats that manipulate the data.

Consider the use of RFID tags embedded within large denomination money notes or mass transit charge cards (such as London’s “Oyster Card” for use on the “tube,” train and bus). Just as with WiFi, attackers can use special high-powered antennas to query RFID tags from a range greater than expected (most RFID technologies are designed to operate within 2cm – 2m ranges, but with a high-powered antenna the attacker may extend this range to 200m). Now imagine that potential attackers are sitting at a second-story window above a busy shopping arcade, using a high-gain antenna to read how much cash people are carrying as they walk by. Similarly, in a not-so-nefarious scenario, consider haggling for the best price with an antiques dealer after he has already read how much cash is in your wallet.

Attacks directed against RFID systems are expected to get more sophisticated as RFID tag usage extends further into our everyday lives. The most serious RFID threats relate to cloning and manipulation of data. Since the chips within RFID tags are very simple, they are trivial to clone — which means that security systems that rely upon stored data such as the unique 96-bit serial number for identification are easy to bypass (e.g., badge access to an office and “smart money” forgery). The most interesting attacks focus on changing the data on a RFID tag so that it affects the system reading it. For instance, consider an RFID tag that stores the owner’s name and address, but has been changed by the attacker to contain classic SQL-Injection strings. The device reads the data, populates a stored procedure with the newly read data and then executes a SQL query of the attacker’s choice — `` ; DROP DATABASE --` springs to mind.



-- Gunter Ollmann, Director of X-Force

Inside the X-Force Engine

Whiro: The ISS 0-day Crawler

For the past few months, X-Force has been working on an exciting new project, “Whiro,” tasked with identifying new 0-day Web browser and binary file exploits, as well as caching potential malcode downloads for further analysis and future preemptive engine development.

The Web browser exploits we are targeting utilize scripting languages within modern browsers to launch their attacks and bypass traditional protection systems. Similarly, with binary file exploits, we are targeting files such as image, sound and movie files that contain exploit material. For example, if Whiro had existed a year ago, we would have observed all the WMF, JPG, GIF, BMP, IMG, etc. exploits in the wild at a very early point.

What makes Whiro unique compared to other Web-crawler projects discussed in the public domain (such as Microsoft’s “Honey-Monkey”) is the inclusion of practical Web browser and shellcode heuristics. Some of the most significant Web browser heuristics are simple enough to integrate into our existing PAM architecture or have already been integrated. Other Web browser heuristics present in Whiro identify the likelihood that a manual investigation should take place. Specifically, this means that interesting JavaScript encodings have been identified that may be obfuscating malicious activities. These more advanced Web browser exploit heuristics are not appropriate for PAM additions (since they require a manual inspection of the HTML), but the Kassel Engineering team has benefited from them while crawling the Web for malicious content — and consequently our Web filtering customers are better protected.

Thus far, we have seen extremely positive results from Whiro, regular Kassel scanning exercises and the initial PAM signatures.

When the Internet Explorer `createTextRange()` 0-day exploit first publicly entered the wild, Whiro had already identified two sites using JavaScript-encoded shellcode and, upon investigation, they were both found to be exploiting this vulnerability.

Kassel’s initial work with the new heuristics technology has already provided multiple leads for further analysis. For example, we are still seeing a lot of sites using the old *MS-ITS* exploit, a few sites using the *JavaProxy COM* object exploit, a few using multiple exploits including the `createControlRange()` exploit (MS05-014) and multiple sites using a commercial off-the-shelf (COTS) exploit module that is updated monthly to include the latest Web browser exploits.

With the initial PAM integration and deployment within Proventia® appliances, our Managed Security Services (MSS) division started observing multiple firings of the `Javascript_Shellcode_Detected` signature. Early statistics indicate upwards of one hundred *0-day* identifications a week. What is particularly striking is that this signature and related ones are frequently detecting COTS deployments of the exploit module just mentioned.

The binary shellcode heuristics are even more exciting. With an excellent degree of confidence, we are able to detect a litany of interesting shellcode including all current Metasploit encodings across all supported processors, third-party x86 encodings covering multiple operating systems and *NIX support over multiple processors including PPC OSX. SCO support has even been implemented — though there have not been any public attacks against this operating system for a long time.

In our own research, we have identified other technologies that sound similar to binary shellcode heuristics but in fact rely on simple algorithms for identifying No-Operation instructions for various processors. Today, Whiro is continuing to identify sites that are serving images containing shellcode. The current implementation of the binary shellcode heuristics is optimized to execute at speeds in excess of 3Gbits/s on modern (non-optimized) x86 machines, and it shares the same architecture as the Web browser exploit heuristics. This degree of speed allows us to consider integration into PAM and VPS.

It is projected that within the next few months, the X-Force engineering team will work toward integrating the binary shellcode heuristics into the core PAM engine. At first, this integration will be targeted towards binary content such as images, sounds and movies. Later on, we plan to examine what may be required to enable this technology to protect remote services. In addition, the X-Force engineering team is working on integration plans to enable the heuristics engine to work alongside the VPS engine.

Ultimately, the heuristic technology that drives Whiro will eat away at the remote exploitation landscape. When deployed in tandem with our other protection engines such as buffer overflow exploit prevention (BOEP), our *Ahead of the threat* technology continues to lead the market by example.

Project Codename: Whiro

Whiro is the Maori (Polynesian) lizard-god of the dead, evil and darkness. He lives in the dark misty underworld and is accompanied by a group of evil spirits. He inspires mischievous and evil thoughts in the minds of people.

-- Robert Freeman, X-Force R&D Researcher

Malcode Corner

Hallowed by Thy Name

Today's antivirus industry suffers a nomenclature problem. When a new threat appears, analysts assign it a name by examining the malicious code in detail — typically using the contents of the malcode and a pinch of imagination to derive a unique name. In many cases the names appear to be random, as each vendor tends to use a custom naming convention. Often, during the first hours of an outbreak, multiple vendors use different names for the threat, only converging to one, two, or three names over the following days. This process can lead to very “personal” names for new threats.

Once family names have been broadly established across the industry, the vendors typically work closely together when naming new variants of that family. For instance, most vendors agree on the names Bagle (Troj_Bagle.AD) or Beagle (W32/Beagle.R@mm) for one threat family. Another example would be the Scano/Inor/Areses outbreak in April; the Inor and Areses names fell into disuse after the first few days.

However, in some cases multiple assigned names continue to linger and cause confusion — e.g., the well-publicized February outbreak resulted in names including W32.Blackmal, Worm_Nyxem, W32/MyWife, Worm_Grew, W32/Tearec.A.worm, and others — all for the same malcode. To exacerbate matters, the media seized on Kama Sutra as a more “sexy” and memorable name for this threat, even though that was simply one of the subject lines that could appear on the e-mail messages attempting to spread the malcode.

As signature AV vendors rush to announce new threats, they compete to acquire “mindshare” and publicity by having their name generally accepted. This is analogous to the marketing concept of “First Mover Advantage.” In the early stages of an outbreak, vendors often resist cooperation and name convergence in direct proportion to the severity of the threat, due primarily to issues of mindshare and public relations. In the end, though, the industry tends to converge on one or a very few names, and the media tend to pick a single “sexy” name and run with that.

Unfortunately, the names produced by this process rarely provide much useful information regarding the threat or its effects on IT systems. What does W32/Zotob.C@mm or W32/Mytob.D mean, anyway? How do these threats spread? What do they do? Occasionally, sometimes accidentally, threats receive more descriptive names such as Worm_Nyxem.E or W32.SQLslammer.worm. More often, however, names such as W32.Blackmal or W32/Beagle.R@mm offer little useful information to the uninitiated.

The MITRE Corporation, a large government contractor, has endeavored to reduce the confusion. MITRE created the Common Malware Enumeration (CME) initiative to assign unique names to threats.

The CME initiative enjoys wide support within the antivirus industry. Unfortunately, CME focuses on the needs of the industry and offers little to the general public.

The names that CME assigns take the form of “CME-” followed by a number. For example, the malcode involved in the W32.Blackmal outbreak received the name CME-24. The CME names the entire family of variants, simplifying the threat

“dictionary.” This is particularly useful to the industry in eliminating the need to cross-reference new attacks to multiple names. This in turn helps simplify the process of ensuring customers are provided the protection they need. However, these names rate very low on the “sexiness scale,” thereby inhibiting widespread adoption outside the industry and particularly by the media. They also tell the customer less about the threat than names like Worm_Nyxem, which least identifies it as a worm. CME does, however, provide a standard nomenclature within the industry. If consistently used by the vendors, it alleviates the issue of “are we talking about the same thing?”.

Even with the CME, customers remain on the outside looking in. They often do not know which threat should arouse their concern — or whether Blackmal and Nyxem represent the same threat. Confusion reigns among customers. Vendors field unnecessary questions. Everyone fights a veritable Babel of names while attempting to assess a crowded threat landscape.

Behavioral detectors, on the other hand, infer malicious intent from the *actions* of the threat rather than implementing the tedious analysis AV vendors use. The behavioral approach allows detection of



malcode upon first sight, without time-consuming human analysis, by inventorying the threat's actions, and can provide the basis for automatic generation of an explanatory name for the threat. For ISS, the real adventure begins here. Unfortunately, since behavioral detectors analyze the threat without human intervention, they cannot provide a sexy, human-friendly name.

With some effort, though, behavioral detectors could provide explanatory names. In fact, they could provide names encompassing not just different variants, but different families, going one step beyond the CME. This would simplify the "dictionary" even more. Explanatory names could help both our customers and the industry more than a name like Worm_Nyxem.E.

Armed with our VPS technology and our deployed 0-day detectors, X-Force is in a position to pioneer a malcode naming strategy.

-- Doug Franklin, Senior X-Force Security Engineer

A Month in Review

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

Malcode:

In April, the X-Force alpha-testing platform for the VPS engine (codename Catfish) identified 66 unique 0-day malcode attacks — before any other antivirus vendor. These 0-day threats comprised variants of four malcode families:

W32.PWS.Goldun

Two sub-variants were discovered, and both were verified to be password stealers possessing backdoor as well as rootkit capabilities. They cause confidential information such as user names and passwords to be disclosed, and in the case of the second variant, the infected system is turned into a proxy to relay connections.

- W32.PWS.Goldun.GU
- W32.PWS.Goldun.HP

W32.Worm.Scano

Fifty-nine variants of malicious HTML files were discovered, collectively containing a total of seven unique variants of this family embedded within them.

All embedded files were verified to be mass-mailing worms. They cause e-mail messages composed in Outlook Express to automatically include a malcode attachment, and also cause arbitrary programs to be downloaded onto the system.

This malcode has polymorphic behavior and in this case was able to evade most signature AV. The

worms embedded within HTML files are loaded onto the system via a script. Most AV products are easily evaded by this type of polymorphic functionality, leading to the large number of variants of the malicious HTML files that were discovered. The following seven variants of the embedded worms were identified as:

- W32.Worm.Scano.E
- W32.Worm.Scano.F
- W32.Worm.Scano.G
- W32.Worm.Scano.H
- W32.Worm.Scano.I
- W32.Worm.Scano.J
- W32.Worm.Scano.K

W32.Worm.Kidala

Two variants were discovered. They were both worms based on the Mytob family. They propagate via e-mail, peer-to-peer (P2P), instant messaging (IM) and by exploiting Microsoft vulnerabilities. In addition, they have Internet Relay Chat (IRC) Bot functionality which allows a remote attacker to control the affected machine via IRC.

- W32.Worm.Kidala.B
- Minor variant of above; only difference is name of dropped file.

MyTob

A new MyTob variant was discovered. It was verified to be a mass-mailer, P2P worm, Network worm and an IRC Bot.

Sample Harvesting:

It is worth noting that as part of the X-Force continued strengthening of ISS antivirus, anti-spyware and anti-malware protection, we investigated and added another 11,717 samples to our malcode zoo this month.

Vulnerabilities:

Last month also saw 554 new vulnerabilities disclosed and analyzed by X-Force. This represents a 31 percent increase over the same month last year, and a 66 percent increase over April 2004.

	Reported				
	Critical	High	Medium	Low	Total
Apr	2	88	334	130	554

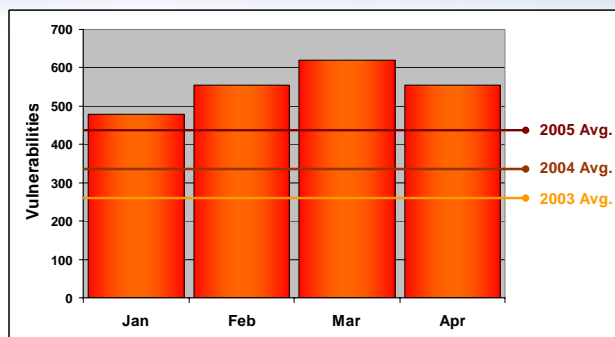
Of these vulnerabilities, 509 were remotely exploitable, 47 could only be exploited locally and 2 could potentially be exploited both remotely and locally.

Vulnerability Breakdown

Bypass Security	17
An attacker can bypass security restrictions such as a firewall or proxy, an IDS system or a virus scanner.	
Data Manipulation	92
An attacker is able to manipulate data stored or used by the host associated with the service or application.	
Denial of Service	79
An attacker can crash or hang a service or system or take down a network.	
File Manipulation	9
An attacker can create, delete, read, modify or overwrite files.	
Gain Access	271
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.	
Gain Privileges	19
Privileges can be gained on the local system only.	
Obtain Information	58
An attacker can obtain information such as file and path names, source code, passwords or server configuration details.	
YTD Total	2207

* Right-hand column represents unique vulnerability count

Vulnerabilities per Month vs. Yearly Average



Top 5 Vulnerable Vendors of the Month

Oracle	36
Ethereal	27
Mozilla	26
Microsoft	20
Apple and Cisco (Joint 5 th Place)	9

* Right-hand column represents unique vulnerability count

X-Force Security Content:

The previous month saw X-Force add 3 new events for various vulnerabilities, deliver one Level-0 X-Press Update® (XPU) related to the Microsoft Super-Tuesday patches, add 8 new blocking responses and add 18 new vulnerability scanner checks.

New Vulnerability Discovery Algorithms:

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

- XML_Subversion_Date_CmdExec
- HTTP_WhatsUp_Login_DoS
- Script_Shell_Command

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.